

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Tesis previa a la obtención del Título de:
Ingeniero de Sistemas**

**TEMA:
“Estudio y Automatización del Proceso de Control y
Monitoreo de Elaboración de Tesis de Estudiantes de la
Carrera de Ingeniería de Sistemas”**

AUTORAS:
Almeida Murillo Zaira
Cárdenas Hernández Roxana Estefanía

DIRECTOR:
Ing. Ricardo Naranjo

Guayaquil, 2013

DEDICATORIA

A Dios por haberme permitido llegar hasta este punto y haberme dado salud, ser el manantial de vida y darme lo necesario para seguir adelante día a día para lograr mis objetivos, metas, además de su infinita bondad y amor.

A mi madre por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor. A mi padre por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

Aquellos que ayudaron directa o indirectamente a realizar este documento.

Roxana Cárdenas

DEDICATORIA

A Dios por estar conmigo en cada paso que doy, por fortalecerme e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte.

A mi mamá Janeth Murillo por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien y por su amor.

A mi padre Hugo Almeida por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por su amor y el valor mostrado para salir adelante.

A mi esposo Iván Orlando por haberme apoyado en toda mi carrera, por su comprensión, amor y haber confiado en que alcanzaría los objetivos trazados.

A mi hija Abby Orlando Almeida porque a pesar de su corta edad a comprendido el tiempo que no le dedique para alcanzar mis metas, por sentirse orgullosa de mi y por su amor que fue y es la mayor motivación para seguir adelante.

Zaira Almeida

AGRADECIMIENTO

A la Universidad Politécnica Salesiana por sus excelentes profesores que me transmitieron conocimientos obtenidos y haberme llevado paso a paso en el aprendizaje.

Al Departamento de Bienestar Estudiantil por brindarme su apoyo y confianza ya que gracias a ellos pude terminar mis estudios.

A todas aquellas personas que con su alentador compromiso fueron importantes en la culminación de este proyecto.

A mi familia por su comprensión y apoyo incondicional.

Ing. Ricardo Naranjo por su gran apoyo y motivación para la culminación de nuestros estudios profesionales, por su apoyo ofrecido en este trabajo, por haberme transmitidos los conocimientos obtenidos.

Roxana Cárdenas

AGRADECIMIENTO

Agradezco a Dios por estar conmigo en cada paso que doy, por darme fuerzas y ayudarme a superar cada obstáculo.

A mi mamá Janeth Murillo por haberme apoyado en todo momento y haberme motivado a seguir adelante aún en los momentos más difíciles inclusive económicos.

A mi padre Hugo Almeida por haberme ayudado en el cuidado de mi hija en mi trayectoria estudiantil y haberme alentado.

A mi esposo Iván Orlando por haberme apoyado incondicionalmente a alcanzar una de mis metas.

A mi hija Abby Orlando Almeida por su amor que me empuja a seguir adelante y ser obediente y comprensiva.

Agradezco a la Universidad Politécnica Salesiana por haberme abierto las puertas, así como al Departamento de Bienestar Estudiantil que estuvo siempre presto a ayudarme y a los docentes que ayudaron en mi formación profesional.

También a mis compañeros que me ayudaron, así como en especial a mi amiga Roxana Cárdenas compañera de Tesis con la cual hemos trabajado como un equipo para sacar adelante este proyecto.

Zaira Almeida

CERTIFICADO

Certifico que el presente trabajo fue realizado por las señoritas Almeida Murillo Zaira Janeth y Cárdenas Hernández Roxana Estefanía, bajo mi supervisión.

Guayaquil, Mayo del 2013

Ing. Ricardo Naranjo
DIRECTOR DE TESIS

DECLARACIÓN DE RESPONSABILIDAD

Nosotras Almeida Murillo Zaira Janeth y Cárdenas Hernández Roxana Estefanía declaramos que el trabajo realizado dentro de este tema de tesis es netamente de nuestra autoría, en lo que corresponde al desarrollo, diseño e implementación.

Guayaquil, Mayo del 2013

Zaira Almeida Murillo

Roxana Cárdenas Hernández

ÍNDICE DE INICIAL

DEDICATORIA	I
AGRADECIMIENTO	III
CERTIFICADO	V
DECLARACIÓN DE RESPONSABILIDAD	VI

ÍNDICE DE CONTENIDOS

CAPITULO 1	- 1 -
1.Diseño De La Investigación.....	- 1 -
1.1 Antecedentes de la Investigación	- 1 -
1.2 Problema de la Investigación	- 1 -
1.2.1 Planteamiento del Problema.....	- 1 -
1.2.2 Formulación del Problema de Investigación.....	- 2 -
1.2.3 Sistematización del Problema de Investigación	- 2 -
1.3 Objetivo de la Investigación.....	- 2 -
1.3.1 Objetivo General	- 2 -
1.3.2 Objetivos Específicos	- 2 -
1.4 Justificación de la Investigación.....	- 3 -
1.5 Marco de referencia de la Investigación	- 3 -
1.5.1 Marco Teórico	- 3 -
1.5.1.1 IDE.....	- 4 -
1.5.1.2 HTML	- 6 -
1.5.1.2.1 Java Script.....	- 8 -
1.5.1.3 Conceptos Web.....	- 11 -
1.5.1.4 Servlet	- 13 -
1.5.1.5 JSP	- 16 -
1.5.1.6 Oracle	- 20 -
1.5.1.7 Modelo Entidad Relacion	- 22 -
1.5.1.9 ENTREVISTA.....	- 31 -
1.5.2 Marco Conceptual	- 31 -
1.6 Formulación De Hipótesis Y Variables	- 33 -

1.6.1 Hipótesis General	- 33 -
1.6.3 Matriz Causa-Efecto.....	- 34 -
1.6.4 Variables	- 35 -
1.7 Aspectos Metodológicos De La Investigación.....	- 35 -
1.7.1 Tipo de estudio.....	- 35 -
1.7.2 Método de investigación.....	- 35 -
1.7.3 Fuentes y técnicas para la recolección de información	- 36 -
1.7.4 Población y Muestra	- 37 -
1.8 Resultados Esperados.....	- 37 -
CAPITULO 2	- 38 -
2. Análisis Presentación De Resultados Y Diagnóstico	- 38 -
2.1 Organigrama de Gestión de Temas de Tesis	- 38 -
2.2 Herramientas de Análisis	- 39 -
2.2.1 Análisis de los Procesos actuales	- 39 -
2.2.1.1 Proceso De La Denuncia de Tesis.....	- 39 -
2.2.1.2 Proceso De Desarrollo De La Tesis.	- 41 -
2.2.1.3 Proceso De Petición de Prorroga.....	- 41 -
2.2.1.4 Proceso De La Presentación Del Trabajo Escrito.	- 42 -
2.2.1.5 Proceso De Declaratoria De Aptitud Para Sustentación... -	42 -
2.2.1.6 Proceso De La Sustentación.	- 44 -
2.2.1.7 Proceso De La Declaratoria De Aptitud Para Titulación . -	44 -
2.3 Análisis estadístico de las encuestas.....	- 45 -
CAPITULO 3	- 51 -
Análisis Y Diseño Del Sistema Propuesto.....	- 51 -
3.1 Arquitectura Del Sistema	- 51 -
3.2 Modelos De Análisis.....	- 52 -
3.2.1 Enfoque de Procesos.....	- 52 -
3.2.2 Diagrama De Procesos Del Sistema	- 54 -
3.2.3 Diagramas De Flujo De Datos	- 58 -
3.2.4 Diagrama De Casos De Uso.....	- 62 -
3.2.5 Diagramas De Estados.....	- 69 -
3.2.6 Diagrama De Secuencia.....	- 72 -
3.2.7 Diagrama De Clases	- 73 -
3.2.8 Análisis Funcional	- 74 -

3.3 Diseño De Arquitectura	- 77 -
3.3.1 Capa De Base De Datos.....	- 77 -
3.3.1.1 Diagrama Relacional	- 77 -
3.3.1.2 Diagrama Entidad - Relación.....	- 79 -
3.3.1.3 Definición de Paquetes	- 101 -
4. CONCLUSIONES.....	- 143 -
5. RECOMENDACIONES	- 144 -
BIBLIOGRAFÍA.....	- 144 -

ÍNDICE DE FIGURAS

Ilustración 1 : Entorno de Desarrollo Eclipse	- 5 -
Ilustración 2: Esquema del servlet	- 14 -
Ilustración 3: Ciclo de Vida de un Software	- 29 -
Ilustración 4: Mapa de Procesos	- 30 -
Ilustración 5: Organigrama de la Gestión de Temas de Tesis	- 38 -
Ilustración 6: Proceso de Denuncia de Tesis	- 40 -
Ilustración 7: Proceso De Desarrollo de la Tesis	- 41 -
Ilustración 8: Proceso de Petición de Prorroga	- 41 -
Ilustración 9: Proceso De La Presentación Del Trabajo Escrito	- 42 -
Ilustración 10: Proceso De Declaratoria De Aptitud Para Sustentación	- 43 -
Ilustración 11: Gráfico Estadístico resultados de la encuesta pregunta 1	- 45 -
Ilustración 12: Gráfico Estadístico resultados de la encuesta pregunta 2	- 46 -
Ilustración 13: Gráfico Estadístico resultados de la encuesta pregunta 3	- 47 -
Ilustración 14: Gráfico Estadístico resultados de la encuesta pregunta 4	- 48 -
Ilustración 15: Gráfico Estadístico resultados de la encuesta pregunta 5	- 49 -
Ilustración 16: Arquitecta MBC	- 51 -
Ilustración 17: Modelo de Procesos	- 52 -
Ilustración 18: Procesos del Sistema	- 54 -
Ilustración 19: Proceso Usuarios	- 55 -
Ilustración 20: Proceso Prorrogas	- 55 -
Ilustración 21: Proceso Monitoreo	- 56 -
Ilustración 22: Proceso Reporte	- 57 -
Ilustración 23: Diagrama de Flujo del Módulo de Usuarios	- 58 -
Ilustración 24: Diagrama de Flujo del Módulo de Usuarios	- 59 -
Ilustración 25: Diagrama de Flujo del Módulo de Prorrogas	- 60 -
Ilustración 26: Diagrama de Flujo del Módulo de Monitoreo	- 61 -
Ilustración 27: Autenticación de usuario	- 63 -
Ilustración 28 : Creación de Usuarios	- 65 -
Ilustración 29: Creación de Prorroga	- 66 -

Ilustración 30: Creación de Monitoreo	- 67 -
Ilustración 31: Creación de Reportes	- 68 -
Ilustración 32: Diagrama de Estado Creación de Usuario	- 70 -
Ilustración 33: Diagrama de Estado Creación de Prorrogas	- 70 -
Ilustración 34: Diagrama de Estado Creación de Monitoreo	- 71 -
Ilustración 35: Diagrama de Secuencia	- 72 -
Ilustración 36: Diagrama de Secuencia	- 73 -
Ilustración 37: Diagrama Relacional del Sistema	- 77 -
Ilustración 38: Diagrama Relacional consultada a la base de la Universidad Politécnica Salesiana	- 78 -
Ilustración 39: Diagrama Entidad Relación del Sistema	- 79 -
Ilustración 40: Diagrama Entidad Relación consultada a la base de la Universidad Politécnica Salesiana	- 80 -
Ilustración 41: Estructura de Paquetes	- 101 -
Ilustración 42: Paquete SEK_OBJ_SMT_AVANCES	- 101 -
Ilustración 43: Paquete SEK_TRX_SMT_AVANCES	- 104 -
Ilustración 44: Paquete SEK_OBJ_SMT_AVANCES	- 106 -

ÍNDICE DE TABLAS

Tabla 1 : Matriz Causa Efecto _____	- 34 -
Tabla 2: Resultado de la Encuesta Pregunta 1 _____	- 45 -
Tabla 3: Resultado de la Encuesta Pregunta 2 _____	- 46 -
Tabla 4: Resultado de la Encuesta Pregunta 3 _____	- 47 -
Tabla 5: Resultado de la Encuesta Pregunta 4 _____	- 48 -
Tabla 6: Resultado de la Encuesta Pregunta 5 _____	- 49 -
Tabla 7: Descripción de Autenticación de usuario _____	- 64 -
Tabla 8: Descripción de Creación de Usuarios _____	- 65 -
Tabla 9: Descripción de Creación de Prorroga _____	- 66 -
Tabla 10: Descripción de Creación de Monitoreo _____	- 67 -
Tabla 11: Descripción Creación de Monitoreo _____	- 69 -
Tabla 12: Detalle del Diagrama de Clases _____	- 74 -
Tabla 13: Tabla de Procedimiento de Usuarios _____	- 75 -
Tabla 14: Tabla de Procedimiento de Prorroga _____	- 75 -
Tabla 15: Tabla de Procedimiento de Reporte _____	- 76 -
Tabla 16 : Tabla de Procedimiento de Monitoreo _____	- 76 -
Tabla 17: Tablas de la base _____	- 83 -
Tabla 18: Tabla de Usuarios _____	- 84 -
Tabla 19: Tabla de Opciones del Menú _____	- 84 -
Tabla 20 : Tabla de Perfil _____	- 85 -
Tabla 21: Tabla de Prorrogas _____	- 85 -
Tabla 22: Tabla de Perfil para Opciones del Menú _____	- 86 -
Tabla 23: Tabla de Capítulos _____	- 86 -
Tabla 24: Tabla de Subcapítulos _____	- 86 -
Tabla 25: Tabla de Avances _____	- 87 -
Tabla 26: Tabla de Tesis _____	- 87 -
Tabla 27: Tabla de Periodos _____	- 88 -
Tabla 28: Tabla de Plazos _____	- 88 -
Tabla 29: Tabla de SNA_RESOLUCION_TRABAJO_GRADO _____	- 90 -

Tabla 30: <i>Tabla SNA_DET_RES_TEM_TES</i>	- 92 -
Tabla 31: <i>Tabla de Sna_Carrera_Institucion</i>	- 92 -
Tabla 32: <i>Tabla de Cliente Local</i>	- 94 -
Tabla 33: <i>Tabla de RH_PERSON</i>	- 97 -
Tabla 34: <i>Tabla SNA_RESOLUCION_TEMA_TESIS</i>	- 98 -
Tabla 35: <i>Tabla de RH_PERSON</i>	- 99 -
Tabla 36: <i>Tabla de SNA_RESOLUCION_TEMA_TESIS</i>	- 100 -

CAPITULO 1

1.DISEÑO DE LA INVESTIGACIÓN

1.1 Antecedentes de la Investigación

La Universidad Politécnica Salesiana nace oficialmente el 4 de agosto de 1994, como centro de educación superior, con la necesidad de formar un profesional integral, científico, práctico, humano, moral y ético, así mismo vinculado con la sociedad para que ayude a la solución de los grandes problemas sociales. La sede Guayaquil está conformado por el campus centenario y su Edificio Administrativo Villa la Joya localizada en Chambers y 5 de Junio.

En la Universidad Politécnica Salesiana de la ciudad de Guayaquil, se lleva a cabo el proceso de tesis, es decir, un proceso en el cual, los estudiantes al culminar la parte académica de su carrera comienzan con la parte del proceso de tesis, el mismo que lo realizan de manera manual, es decir utilizando hojas de cálculos en Excel y en papel, lo cual no optimiza un proceso al 100%.

Pero no tan solo es un problema para los estudiantes también para el personal administrativo y tutores ya que tienen que darle seguimiento manualmente al proceso.

Los sistemas informáticos en la actualidad ayudan mucho a simplificar el trabajo debido al ahorro de tiempo y recursos.

1.2 Problema de la Investigación

1.2.1 Planteamiento del Problema

En la Universidad Politécnica Salesiana de la ciudad de Guayaquil, se ha observado un representativo número de estudiantes que no cumplen a tiempo sus plazos para la elaboración de su tesis. Los estudiantes no tienen presente la fecha cuando iniciaron su tesis y se descuidan del plazo que tienen para presentarlo, no solicitan oportunamente sus prorrogas y no existen medios automatizados para su control y seguimiento.

En futuros años debido al incremento de los estudiantes que cumplen con los requisitos para realizar el proceso de tesis la cantidad de datos se incrementará aumentando la probabilidad de que ocurran errores y pérdida de información. Así mismo en el momento de realizar las debidas actualizaciones de los datos causará inconvenientes, provocando que las actividades no se cumplan en el tiempo determinado para estudiantes y administrativos.

Una solución viable para la corrección del problema analizado es realizar la automatización de cada uno de los procesos manuales implementándolos en un sistema web que se acople a todas las actividades realizadas en este proceso.

1.2.2 Formulación del Problema de Investigación

¿Qué factores inciden para que algunos estudiantes no cumplan a tiempo sus plazos para la elaboración de tesis?

1.2.3 Sistematización del Problema de Investigación

¿Qué consecuencia tiene que los estudiantes se atrasen al no entregar a tiempo la elaboración de la tesis?

¿En qué medida el proceso de control y seguimiento de tesis demanda tiempo?

¿Qué incidencias tiene el llevar a cabo el proceso manual y no automatizado?

1.3 Objetivo de la Investigación

1.3.1 Objetivo General

Identificar los problemas que inciden para que algunos estudiantes no cumplan a tiempo sus plazos para la elaboración de tesis.

1.3.2 Objetivos Específicos

Identificar las consecuencias que causaría a los estudiantes al no entregar a tiempo la elaboración de la tesis.

Identificar en qué medida demanda tiempo el control y seguimiento de la tesis.

Determinar las incidencias al llevar a cabo el proceso manual y no automatizado.

1.4 Justificación de la Investigación

El proyecto genera otra alternativa a los estudiantes egresados de la Universidad Politécnica Salesiana para los que no tienen tiempo de poder acercarse a la institución o no recuerdan la fecha cuando iniciaron su tesis y se descuidan en el plazo que tienen para presentar, esta alternativa automatizara todo el proceso de tesis para el control y seguimiento de la misma.

La persona encargada de llevar el proceso manualmente tiene que verificar cual es la fecha próxima a sustentar del estudiante egresado para así poder comunicarse con el estudiante ya sea mediante vía telefónica o email, este proceso les lleva mucho tiempo interrumpiendo sus labores diarias.

La presente investigación, análisis, diseño e implementación del proyecto busca dar solución al problema planteado.

El sistema de recordatorios será parametrizable para así por medio de correo electrónico recordarles a los estudiantes egresados cuando es el vencimiento de su tema de tesis y se puedan acercar en la fecha indicada a pedir prorroga si es necesario.

El proyecto está orientado netamente a la Universidad Politécnica Salesiana sede de Guayaquil Carrera de Ingeniería de Sistemas y podrá ser configurado según la necesidad de la misma.

1.5 Marco de referencia de la Investigación

1.5.1 Marco Teórico

Se analizaran todos los aspectos a tomar en cuenta en este desarrollo y así hacer una correcta implementación minuciosa, analizando y estandarizando a efectos de asegurar y dar un marco seguro y confiable que garanticen la calidad del producto final.

1.5.1.1 IDE

¿QUÉ ES UN IDE?

“**Entorno de desarrollo integrado**, llamado también **IDE** (sigla en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios. IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.”^[1]

COMPONENTES

- Un editor de texto
- Un compilador
- Un intérprete
- Un depurador
- Un cliente
- Posibilidad de ofrecer un sistema de control de versiones.
- Factibilidad para ayuda en la construcción de interfaces gráficas de usuario.

LENGUAJES

Algunos entornos son compatibles con múltiples lenguajes de programación, como Eclipse o NetBeans, ambos basados en Java o MonoDevelop, basado en C#. También

¹ (DEITEL & DEITEL, 2004)

puede incorporarse la funcionalidad para lenguajes alternativos mediante el uso de plugins. Por ejemplo, Eclipse y NetBeans tienen plugins para C, C++, Ada, Perl, Python, Ruby y PHP, entre otros.

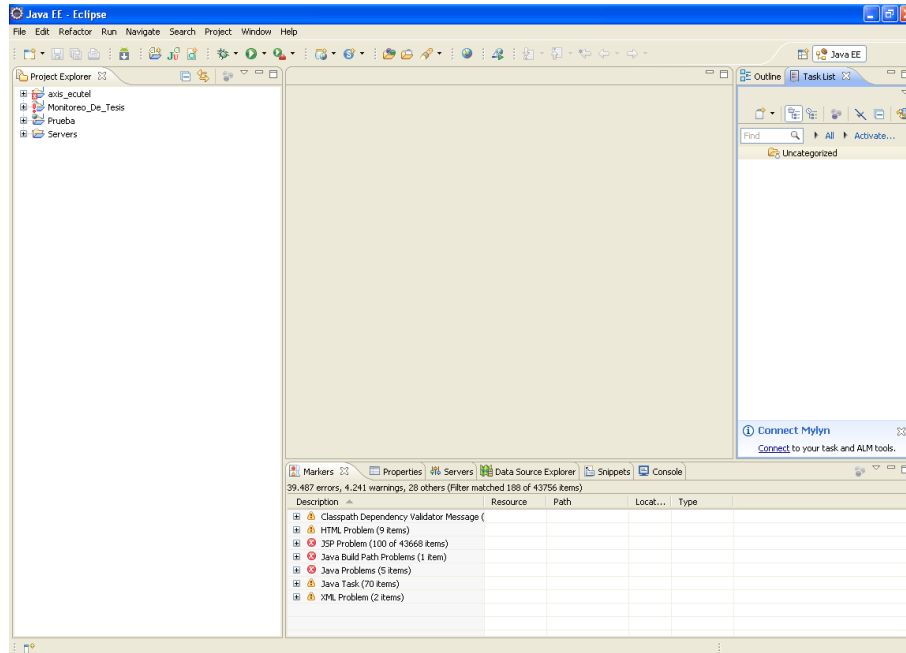


Ilustración 1 : Entorno de Desarrollo Eclipse

Fuente: Los Autores

1.5.1.1.1 Eclipse

“Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.”^[2]

Arquitectura de Eclipse

La base para Eclipse es la Plataforma de cliente enriquecido (del Inglés Rich Client Platform RCP).

Componentes

Los siguientes componentes constituyen la plataforma de cliente enriquecido:

- Plataforma principal - inicio de Eclipse, ejecución de plugins
- OSGi - una plataforma para bundling estándar.
- El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- JFace - manejo de archivos, manejo de texto, editores de texto
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes

Características

Es un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (*wizards*) para creación de proyectos, clases, etc., y refactorización.

1.5.1.2 HTML

¿QUÉ ES HTML?

“Html es el lenguaje con el que se definen las paginas web. Basicamente se trata de un conjunto de etiquetas que sirven para definir la forma en la que presentar texto y otros elementos de la pagina. Html es el lenguaje con el que se escriben las paginas web. Las paginas web pueden ser vistas por el usuario mediante un tipo de aplicaciones llamada navegador. Podemos decir por lo tanto que el html es el lenguaje usado por los

² Wikipedia.”Eclipse”. http://es.wikipedia.org/wiki/Eclipse_%28software%29

navegadores para mostrar las paginas web al usuario, siendo hoy en dia la interface mas extendida en la red.”^[3]

ARQUITECTURA HTML

Un documento está formado por elementos:

- Títulos
- Párrafos
- Listas
- tablas

Para delimitar los elementos se utilizan etiquetas :

```
<nombre-etiqueta>  
elemento  
</nombre-etiqueta>
```

Ejemplo:

```
<TITLE>Un título</TITLE>  
ABRE ETIQUETA  
CIERRA ETIQUETA
```

Los contenidos se encuentran anidados unos dentro de otros en etiquetas predeterminadas del propio lenguaje que debemos abrir y cerrar para el correcto visionado.

TAGS BÁSICOS - ESTRUCTURA BÁSICA

```
<HTML>  
<HEAD></HEAD>  
<TITLE>Mi nombre es José</TITLE>  
<BODY>  
    Soy De Valencia  
</BODY>  
</HTML>
```

³ HTML: Milton Orellana. Primera edición_2004.

TAGS BÁSICOS / DOCUMENTO SIMPLE

`<html>`

`<head>`

`<TITLE>Un documento simple</TITLE>`

`</head>`

`<body>`

Esto es un texto que no tendrá estilos, ni características de ningún tipo.

`</body>`

`</head>`

`</html>`

1.5.1.2.1 JAVA SCRIPT

“JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador... aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor.

JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.”^[4]

¿DÓNDE PUEDO VER FUNCIONANDO JAVASCRIPT?

Entre los diferentes servicios que se encuentran realizados con Javascript en Internet se encuentran:

- Correo
- Chat
- Buscadores de Información

También podemos encontrar o crear códigos para insertarlos en las páginas como:

- Reloj
- Contadores de visitas
- Fechas
- Calculadoras
- Validadores de formularios
- Detectores de navegadores e idiomas

CARACTERÍSTICAS

Su sintaxis es similar a la usada en Java y C, al ser un lenguaje del lado del cliente este es interpretado por el navegador, no se necesita tener instalado ningún Framework.

- Variables: `var = "Hola", n=103`
- Condiciones: `if(i<10){ ... }`
- Ciclos: `for(i; i<10; i++){ ... }`
- Arreglos: `var miArreglo = new Array("12", "77", "5")`
- Funciones: Propias del lenguaje y predefinidas por los usuarios
- Comentarios para una sola línea: `// Comentarios`

⁴ (CASTRO, 2007)

- Comentarios para varias líneas:

```
/*  
Comentarios  
*/
```

- Permite la programación orientada a objetos.
- Las variables pueden ser definidas como: string, integer, flota, boolean simplemente utilizando “var”. Podemos usar “+” para concatenar cadenas y variables.

SINTAXIS

- No es necesario declararlas
- Se emplea la palabra reservada var
- Para las variables letras , numeros o_.
- Primer carácter: no número
- Variable $\neq$ Palabra reservada
- Valor inicial null

MÉTODOS DE PETICIÓN

La primera línea de una petición contiene los comandos HTTP, conocidos como métodos. Existen varios, pero los más conocidos y utilizados son tres: GET, HEAD y POST.

El método GET se utiliza para recuperar información identificada por un URI por parte de los navegadores. El método GET también se puede utilizar para pasar una pequeña cantidad de información al servidor en forma de pares atributo-valor añadidos al final del URI detrás de un símbolo de interrogación, ?.

La longitud de la petición GET está limitada por el espacio libre en los buffers de entrada. Por lo que para mandar una gran cantidad de información al servidor ha de utilizarse el método POST.

El método POST se refiere normalmente a la invocación de procesos que generan datos que serán devueltos como respuesta a la petición. Además se utiliza para aportar datos

de entrada a esos programas. En este caso los pares atributo-valor son incluidos en el cuerpo de la petición separados por ampersand.

EJEMPLO:

```
<form action="post">
  <input type="text" name="email">
  <input type="submit" name="boton" value="enviar">
</form>
```

De este modo el método POST no sufre de las limitaciones de espacio y puede enviar mucha más información al servidor.

1.5.1.3 CONCEPTOS WEB

INTRANET

“Una intranet es una red de ordenadores privados que utiliza tecnología internet para compartir dentro de una organización parte de sus sistemas de información y sistemas operacionales.

Tiene como función principal proveer lógica de negocios para aplicaciones de captura, informes y consultas con el fin de facilitar la producción de la organización.”^[5]

INTERNET

“Internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.”^[6]

PROTOCOLO TCP/IP

“Un protocolo es una convención o estándar que controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales.

⁵ (LOPEZ QUIJADO, 2008)

⁶ (LOPEZ QUIJADO, 2008)

Los protocolos pueden ser implementados por hardware, software, o una combinación de ambos. A su más bajo nivel, un protocolo define el comportamiento de una conexión de hardware.”^[7]

TCP

Transmission Control Protocol (en español Protocolo de Control de Transmisión) o TCP, es uno de los protocolos fundamentales en Internet. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto (Multiplexión de servicios).

PROTOCOLO HTTP

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.

URL

“El URL es la cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en la Internet. Existe un URL único para cada página de cada uno de los documentos de la World Wide Web.”^[8]

⁷ (LOPEZ QUIJADO, 2008)

⁸ (LOPEZ QUIJADO, 2008)

CLIENTES Y SERVIDORES.

CLIENTES.

Es un equipo, proceso, programa, aplicación, sistema, que accede a recursos y servicios brindados por otro llamado servidor, generalmente de forma remota.

SERVIDORES.

Es un equipo, aplicación, programa, sistema que realiza algunas tareas en beneficio de otras aplicaciones. Este es el significado original del término, es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor.

En conclusión un servidor también puede ser un proceso que entrega información o sirve a otro proceso. El modelo CLIENTE-SERVIDOR no necesariamente implica tener dos ordenadores, ya que un proceso cliente puede solicitar algo como una impresión a un proceso servidor en un mismo ordenador.

1.5.1.4 Servlet

“Los servlets son objetos que corren dentro y fuera del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad.

La palabra *servlet* deriva de otra anterior, *applet*, que se refería a pequeños programas que se ejecutan en el contexto de un navegador web. Por contraposición.

El uso más común de los *servlets* es generar todas páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.”^[9]

Un Servlet es una clase Java usada para extender la capacidad de las aplicaciones basadas en el modelo cliente servidor y ejecución petición respuesta. Los Servlets son una potente herramienta para la generación de contenido dinámico en la Web.

⁹ (DEITEL & DEITEL, 2004)

Ciclo de Vida

El ciclo de vida de un Servlet se divide en los siguientes puntos:

1. El cliente solicita una petición a un servidor vía URL.
2. El servidor recibe la petición.
 - Si es la primera, se utiliza el motor de Servlets para cargarlo y se llama al método `init()`.
 - Si ya está iniciado, cualquier petición se convierte en un nuevo hilo. Un Servlet puede manejar múltiples peticiones de clientes.
3. Se llama al método `service()` para procesar la petición devolviendo el resultado al cliente.
4. Cuando se apaga el motor de un Servlet se llama al método `destroy()`, que lo destruye y libera los recursos abiertos.

Esquema

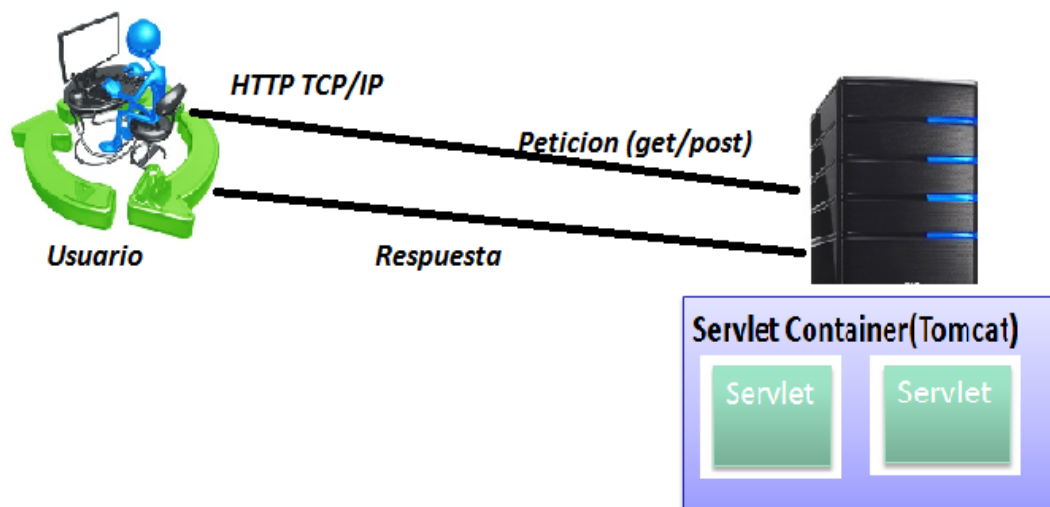


Ilustración 2: Esquema del servlet

Fuente: Los Autores

Características

- Los Servlets inician un nuevo thread, hilo, (más que un nuevo proceso) con cada petición en vez de crear un nuevo objeto en memoria.
- Portabilidad: Los servlets son tan portables como cualquier otra aplicación de Java.
- Seguridad: Los lenguajes compilados como Java (o C) proveen mejor seguridad que los lenguajes que interpretan scripts.
- Desarrollo: Hay IDEs disponibles desde Sun, Microsoft, Symantec y muchas otras, proporcionando simples entornos GUI. Las IDEs pueden proveer útiles herramientas fáciles de aprender para nuevos desarrolladores.

Ventajas

- Rendimiento: cada petición es procesada por un único proceso en el contenedor de servlets.
- Portabilidad: heredado de Java.
- Rápido desarrollo: acceso a las riquísimas librerías de Java.
- Robustez: gestionados por la máquina virtual de Java (garbage collection).
- Amplio soporte: muchos desarrolladores y compañías utilizan esta tecnología.

Ejemplo

Código de ejemplo de un Servlet que procesa una petición GET y devuelve una página web HTML sencilla:

```

package org.pruebas;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HolaMundoServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0
Transitional//EN\">");
        out.println("<html>");
        out.println("<head> <title>Ejemplo Hola Mundo</title> </head>");
        out.println("<body>");
        out.println("<h1>¡Hola Mundo!</h1>");
        out.println("</body> </html>");
    }
}

```

1.5.1.5 JSP

“JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Esta tecnología es un desarrollo de la compañía Sun Microsystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1.

Las JSP's permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.”^[10]

¹⁰ (DEITEL & DEITEL, 2004)

Arquitectura

JSP puede considerarse como una manera alternativa, y simplificada, de construir servlets. Cada versión de la especificación de JSP está fuertemente vinculada a una versión en particular de la especificación de servlets.

El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario.

JSP -> Servidor Aplicaciones (Servlets) -> Cliente (Navegador)

Es posible enriquecer el lenguaje de etiquetas utilizado por JSP. Para ello debemos extender la capa de alto nivel JSP mediante la implementación de Bibliotecas de Etiquetas (Tags Libraries).

TagLibs -> JSP -> Servidor Aplicaciones (Servlets) -> Cliente (Navegador)

El rendimiento de una página JSP es el mismo que tendría el servlet equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

Ventajas

- Separación de datos estáticos/dinámicos.
- Independencia de formato/plataforma.
- Sencillez (sabiendo servlets).

Ciclo de Vida

- No representan una nueva tecnología:
- Semánticamente equivalen a los servlets.
- Sólo facilitan el desarrollo y mantenimiento.

- Ciclo de vida:
- El contenedor de servlets traduce la página JSP a su equivalente servlet la primera vez que se accede.
- El servlet es compilado y cargado en memoria siguiendo su ciclo de vida convencional.
- Si la página es modificada, el contenedor detecta el cambio y vuelve a repetir el proceso.

Elementos

- extensión .jsp.
- Aunque el código parezca más bien HTML, el servidor lo traduce a un servlet en la primera petición.
- elementos en un JSP:
 - ✓ Elementos script (scriptlets)
 - ✓ Directivas
 - ✓ Acciones

INTRODUCCION DE CODIGO

EXPRESION

`<%= expression %>;`

La Expresión es evaluada y situada en la salida.

El equivalente XML es

`<jsp:expression> expression </jsp:expression>`

Las variables predefinidas son request, response, out, session, application, config, y pageContext.

Scriptlet JSP

`<% code %>;`

El código se inserta en el método service.

El equivalente XML es:

```
<jsp:scriptlet>code</jsp:scriptlet>
```

EJEMPLO DE CÓDIGO DE UNA PÁGINA JSP:

```
<%@ page errorPage="myerror.jsp" %>
<%@ page import="com.foo.bar" %>
<html>
<head>
<%! int serverInstanceVariable = 1;%>
...
<% int localStackBasedVariable = 1; %>
```

EJEMPLO DE UNA COMPILACIÓN O "SALIDA" JSP:

```
package jsp_servlet;
import java.util.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
import com.foo.bar; //importado como resultado de <%@ page import="com.foo.bar" %>
import ...

class _myservlet implements javax.servlet.Servlet, javax.servlet.jsp.HttpJspPage {
    //insertado como
    //resultado de <%! int serverInstanceVariable = 1;%>
    int serverInstanceVariable = 1;
    ...
    public void _jspService( javax.servlet.http.HttpServletRequest request,
                             javax.servlet.http.HttpServletResponse response )
        throws javax.servlet.ServletException,
               java.io.IOException
```

```

{
    javax.servlet.ServletConfig config = ...; //obtener la configuración del servlet
    Object page = this;
    PageContext pageContext = ...; //obtener el contexto de la página para esta petición
    javax.servlet.jsp.JspWriter out = pageContext.getOut();
    HttpSession session = request.getSession( true );

```

Para ejecutar las páginas JSP, se necesita un servidor Web con un contenedor Web que cumpla con las especificaciones de JSP y de Servlet. Tomcat es una completa implementación de referencia para las especificaciones Java Servlet y JSP.

1.5.1.6 Oracle

“Oracle es un sistema de gestión de base de datos objeto-relacional (o ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation”.

Es una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial el cual es utilizado en empresas grandes debido a su costo elevado, debido a esta característica no es utilizado como otras bases de datos. Las herramientas de desarrollo que utiliza Oracle son Oracle Designer y Oracle Developer las cuales permiten realizar la programación.

Estructura física y lógica

Las estructuras físicas son almacenadas en dispositivos tangibles, cada archivo tiene un espacio en el sistema operativo y varios archivos conforman su estructura física.

A la estructura lógica le corresponde un espacio por unidad, pero sus limitaciones son independientes de las localizaciones de espacio físico.

Programas y archivos que componen Oracle

- Un Sistema administrador de bases de datos relacionales RDBMS Oracle está compuesto por tres partes principales, que son:

El kernel

El Kernel es el núcleo de Oracle, el cual maneja las siguientes tareas:

- Manejar el almacenamiento y definición de los datos.
- Suministrar y limitar el acceso a los datos y la concurrencia de los usuarios.
- Permitir los backup y la recuperación de los datos.
- Interpretar el SQL y PL/SQL.

El Kernel es cargado a la memoria al inicio de las operaciones y es usado por cada base de datos existente en el equipo.

El PGA (Programa Global Área)

Este componente consta de datos e información de control de los procesos, asegurando el uso correcto. Su función es contener información acerca de las conexiones y los procesos que se realizan en Oracle, su tamaño es variable pero no dinámico y se activa al conectarse un usuario.

El SGA (System Global Area)

Su función es definir buffers a través de los cuales las transacciones y el almacenamiento de dato fluyen. El SGA es localizado en memoria al iniciarse una instancia y desaparece al bajarla. Su tamaño no puede ser cambiado, pero si puede ser visto. Su longitud está definida por los parámetros del archivo de iniciación INIT.ORA.

Su estructura está Compuesta por:

- Diccionario Cache
- Los Redo Log Buffers
- Los Database Buffers

Diccionario cache

Es un conjunto de tablas que son usadas para proveer información asociada con la base de datos, donde se definen los nombres de usuarios, privilegios, objetos, retenciones y espacios que conforman un RDBMS Oracle.

Redo log buffers:

Es un espacio reservado en memoria que contiene los cambios realizados a los bloques de datos, permitiendo la reconstrucción de la base de datos en caso de ocurrir un accidente.

Database buffers:

Es un espacio reservado en memoria para las operaciones normales de la base de datos, el cual depende del tamaño especificado en el archivo de inicialización (INIT.ORA).

Ventajas de usar Oracle

- Oracle es un motor de base de datos usado a nivel mundial.
- Puede ejecutarse en todas las plataformas.
- Soporta Funciones de diseño de bases de datos muy completo permitiendo ejecutar triggers y procedimientos almacenados de una forma muy potente.
- Permite el uso de particiones para la mejor eficiencia.
- Puede ejecutarse en múltiples sistemas operativos.
- Tiene una estructura de datos compleja.
- Oracle es la base de datos usada en la web.
- Tiene un soporte aceptable.

1.5.1.7 MODELO ENTIDAD RELACION

“Un diagrama o modelo entidad-relación (a veces denominado por sus siglas del inglés, *E-R* "Entity relationship", o del español *DER* "Diagrama de Entidad Relación") es una

herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.”^[11]

Entidades y Relaciones

En el modelo E/R se parte de una situación real a partir de la cual se definen entidades y relaciones entre dichas entidades:

Entidad

Objeto del mundo real sobre el que queremos almacenar información (Ej: una persona).

Las entidades están compuestas de:

Atributos.

Son los datos que definen el objeto (para la entidad persona serían DNI, nombre, apellidos, dirección,...). De entre los atributos habrá uno o un conjunto de ellos que no se repite; a este atributo o conjunto de atributos se le llama clave de la entidad, (para la entidad persona una clave sería DNI). En toda entidad siempre hay al menos una clave que en el peor de los casos estará formada por todos los atributos de la tabla. Ya que puede haber varias claves y necesitamos elegir una, lo haremos atendiendo a estas normas:

- Que sea única.
- Que se tenga pleno conocimiento de ella.- ¿Por qué en las empresas se asigna a cada cliente un número de cliente?.
- Que sea mínima, ya que será muy utilizada por el gestor de base de datos.

Relación

Asociación entre entidades, sin existencia propia en el mundo real que estamos modelando, pero necesaria para reflejar las interacciones existentes entre entidades. Las relaciones pueden ser de tres tipos:

¹¹ (PEREZ, 2008)

Relaciones 1-1.- Las entidades que intervienen en la relación se asocian una a una (Ej: la entidad HOMBRE, la entidad MUJER y entre ellos la relación MATRIMONIO).

Relaciones 1-n.- Una ocurrencia de una entidad está asociada con muchas (n) de otra (Ej: la entidad EMPERSA, la entidad TRABAJADOR y entre ellos la relación TRABAJAR-EN).

Relaciones n-n.-Cada ocurrencia, en cualquiera de las dos entidades de la relación, puede estar asociada con muchas (n) de la otra y viceversa (Ej: la entidad ALUMNO, la entidad EMPRESA y entre ellos la relación MATRÍCULA).

Restricciones

Son reglas que deben mantener los datos almacenados en la base de datos.

Cardinalidad de las Relaciones

El tipo de cardinalidad se representa mediante una etiqueta en el exterior de la relación, respectivamente: "1:1", "1:N" y "N:M", aunque la notación depende del lenguaje utilizado, la que más se usa actualmente es el unificado. Otra forma de expresar la cardinalidad es situando un símbolo cerca de la línea que conecta una entidad con una relación:

"0" :si cada instancia de la entidad no está obligada a participar en la relación.

"1" :si toda instancia de la entidad está obligada a participar en la relación y, además, solamente participa una vez.

"N" , "M", ó "*" : si cada instancia de la entidad no está obligada a participar en la relación y puede hacerlo cualquier número de veces.

Ejemplos de relaciones que expresan cardinalidad:

Cada esposo (entidad) está casado (relación) con una única esposa (entidad) y viceversa. Es una relación 1:1.

Una factura (entidad) se emite (relación) a una persona (entidad) y sólo una, pero una persona puede tener varias facturas emitidas a su nombre. Todas las facturas se emiten a nombre de alguien. Es una relación 1:N.

Un cliente (entidad) puede comprar (relación) varios artículos (entidad) y un artículo puede ser comprado por varios clientes distintos. Es una relación N:M.

Restricciones de llave

Relación “Trabaja_en”:

- Un empleado puede trabajar en un departamento
- Un departamento puede tener varios empleados
- Sin embargo, cada departamento puede tener a lo más un jefe por la restricción de llave de la relación administra

Restricciones estructurales

Es una notación alternativa a las restricciones de llave (cardinalidad) que incluye un par de números enteros (mín, máx) a cada participación.

Restricciones de participación

La existencia de una entidad depende de que esté relacionado con otra entidad a través de un tipo de vínculo.

Claves

Al menos debe existir una clave .Tipos de claves:

- Principal o primaria
- Secundarias a alternas
- Foráneas o externas
- Simples
- Compuestas

1.5.1.8 SISTEMA INFORMÁTICO AUTOMATIZADO

SISTEMA

Conjunto de dos o más elementos que se encuentran relacionados entre sí, pueden dividirse en partes. Estos subsistemas son sistemas más pequeños que tienen las siguientes propiedades:

- Cada subsistema realiza una acción sobre el sistema.
- El subsistema no puede ser descompuesto en subconjuntos independientes.
- Un sistema de información ayuda a la toma de decisiones y en la solución de problemas dentro de una organización.

SISTEMA DE INFORMACIÓN

El sistema de información producen funciones como: clasificación, transferencia, almacenamiento, recuperación, evolución.

Su función es proporcionar información para una buena coordinación y toma de decisiones. Incluyendo componentes para realizar estos procesos automáticos.

El objetivo de un sistema de información es proporciona la información necesaria para la solución de problemas y tomar decisiones dentro de una organización y que de esta forma permitan controlar y dirigir su efectividad, acción y utilización.

Un sistema de información es un proceso a través del cual se recopilan, clasifican, procesan, interpretan y se sintetiza cantidades de información, con el objetivo de obtener conclusiones, que se informarán a los jefes inmediatos y personas que pertenecen a la organización para alinear la toma de las decisiones.

Existen diferentes tipos de sistemas de Información se diferencian porque sus datos son entradas y salidas, tipo de procesamiento que ejecutan y la composición de su estructura. El objetivo del sistema define los elementos que contendrá y así mismo por las necesidades de la organización definirá sus funciones.

Para comprender el funcionamiento de un sistema se debe tener en cuenta las funciones que este ejecuta:

Procesamiento de Transacciones: Consiste en recolectar, clasificar, ordenar, calcular, reducir y almacenar los datos originados por las transacciones que se realizan de acuerdo a las actividades en la organización.

Almacenamiento de Archivos: Consiste en recopilar la información acumulada por el procesamiento de transacciones de acuerdo a una estructura de almacenamiento definida las cuales pueden ser una base de datos o archivos buscando que facilite su almacenamiento, actualización y acceso.

Mantenimiento de Archivos: El conjunto de registros o base de datos del sistema deben mantenerse actualizados. Para esto se deben de ejecutar el respectivo mantenimiento que incluye acciones como el ingreso, la modificación y la eliminación de datos en los medios de almacenamiento.

Generación de Reportes: Es una de las funciones esenciales en los sistemas ayudando a obtener resultados inmediatos de la información requerida almacenada en una base de datos y transmitirla a las personas que la necesiten dentro de la organización o fuera de ella.

Ciclo de Vida de un Sistema Informático

El ciclo de vida de desarrollo de sistemas informáticos puede dividirse en actividades o fases, dado que existe gran cantidad de variantes que dependen de la organización, del tipo de sistema que se realizará.

Para el desarrollo de un sistema se necesita hacerlo con eficiencia ya que involucra actividades como análisis, diseño, planificación actividades que deben planificarse para poder organizar el trabajo y garanticen el trabajo. El ciclo clásico de un sistema Informático consta de las siguientes etapas:

Reconocimiento del problema

Surge cuando el usuario reconoce que tiene problemas con los medios con que cuenta actualmente para llevar a cabo su trabajo. Así comienza esta fase que trata de reemplazar el sistema existente por otro. En esta fase interviene totalmente el usuario.

Estudio de la factibilidad

Se decide si el usuario necesita o no una computadora. Este estudio sirve para:

- Identificar los problemas con el sistema actual.
- Identificar el alcance del sistema a ser estudiado.
- Identificar los principales objetivos del nuevo sistema.
- Identificar un número de soluciones que pueden satisfacer las necesidades del usuario dentro de su esquema.
- Desarrollar estimados de los beneficios y desventajas de cada solución.
- Desarrollar esquemas de cómo puede llevarse a cabo el proyecto teniendo una idea de los recursos que se requieren.
- Obtener puntos de vista del usuario y el administrador sobre las Modificaciones.
- Obtener una decisión de si se lleva a cabo la parte de análisis.

Todo este estudio evitará el gasto de un análisis de un proyecto imposible. En él intervienen el usuario y el analista.

Análisis

Es la fase de diseño externo. Consiste en cuestionar al usuario sobre qué hace el sistema, qué características extras él quiere en su nuevo sistema y qué restricciones debe satisfacer. La salida del análisis debe incluir una especificación funcional y un análisis estructurado que contiene los requerimientos para el nuevo sistema, los cuales el usuario debe leer, analizar y señalar lo que él quiere.

Diseño

Es la fase de diseño interno. Consiste en definir cómo organizar lo anterior de forma adecuada para la ejecución. Incluye la realización de diagramas de estructura, explicaciones del programa, etc. (diseño preliminar). Posteriormente se lleva a cabo un diseño detallado donde se describen las especificaciones de los módulos.

Implementación

Es la fase de programación o escritura del código. Lo que se produce en el diseño se lleva a código.

Prueba

En esta etapa se planea el diseño de casos de prueba con el fin de asegurar la funcionalidad del programa.

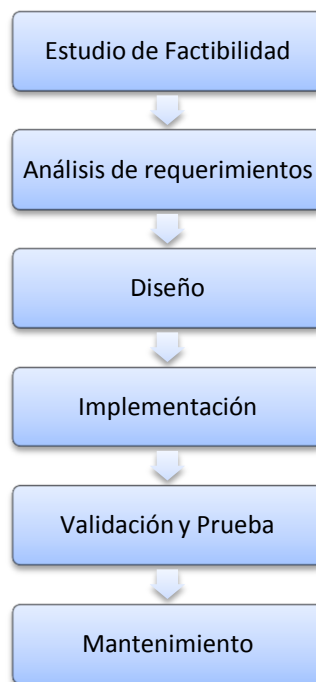


Ilustración 3: Ciclo de Vida de un Software

Fuente: Los Autores

Mapa de procesos

Un Mapa de Procesos de una Organización, es la representación gráfica de los procesos de ésta y de sus interrelaciones. A la hora de realizar el Mapa de Procesos, habrá que identificar primero todos los procesos que integran la organización y posteriormente se deberán clasificar.

En el Mapa se deberán identificar tres tipos de procesos:

- Procesos estratégicos: Son todos los procesos que proporcionan pautas de acción para todos los demás procesos y son realizados por la dirección general o en nombre de esta. Se suelen hacer referencia a reglamentación, leyes, normativas,... aplicables al producto o servicio y que no son controladas por el mismo.
- Procesos clave u operativos: hacen referencia a los Procesos de la cadena de Valor de la Organización y tienen impacto en el cliente creando valor para éste. Son las actividades esenciales de la Organización, su razón de ser.
- Procesos de apoyo o soporte: son aquellos que dan apoyo a los procesos fundamentales de la Organización.

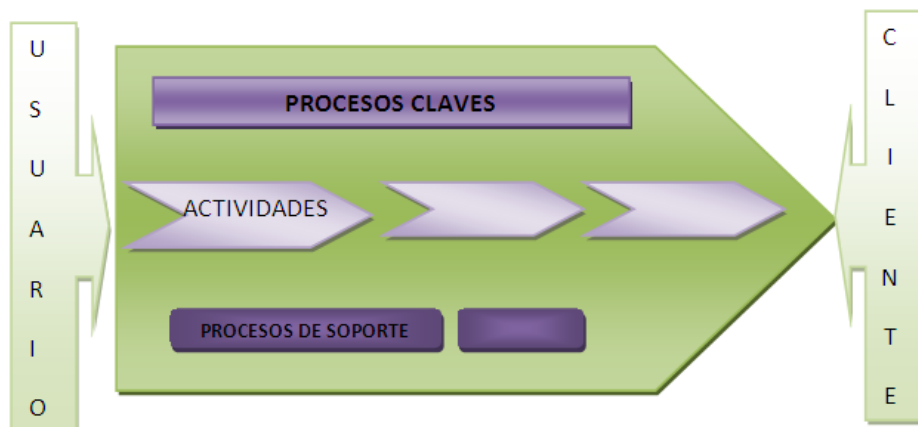


Ilustración 4: Mapa de Procesos

Fuente: Las Autoras

1.5.1.9 ENTREVISTA

Es una conversación en la que se usa el formato de preguntas y respuestas enfocadas en un propósito específico, con esto se pretenderá obtener la opinión actual de los involucrados en la manera como se lleva el proceso actualmente.

A través de esta herramienta se pueden obtener importantes conclusiones, incluso más que la observación de los hechos.

En una entrevista se debe seguir los siguientes pasos:

- Conocer los antecedentes.
- Determinar los objetivos de la entrevista.
- Definir el personal que va a ser entrevistado.
- Definir el tipo de preguntas y la estructura.

1.5.2 Marco Conceptual

Base de Datos: Es un “almacén de datos” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

MER: Modelo Entidad Relación, es una herramienta para el modelado de datos que expresan interrelaciones y propiedades dentro de un sistema de información.

Modularidad: Unión de varias partes de un sistema que interactúan entre sí y que trabajan para cumplir actividades y alcanzar un objetivo.

Sistema de Información: Procesos asociados con el objetivo de proporcionar la información necesaria para la solución de problemas y tomar decisiones dentro de una organización.

CSS: El nombre **hojas de estilo en cascada** viene del inglés *Cascading Style Sheets*, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

HTML: Siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

SQL: El Lenguaje de consulta estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

FRAMEWORK: Un framework, en el desarrollo de software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

ACCESIBILIDAD: La accesibilidad web se refiere a la capacidad de acceso a la Web y a sus contenidos por todas las personas independientemente de la discapacidad (física, intelectual o técnica) que presenten o de las que se deriven del contexto de uso (tecnológico o ambiental). Esta cualidad está íntimamente relacionada con la usabilidad.

E-mail: Forma de correo por Internet que permite el envío de mensajes a otra persona conectada a la Red. En los últimos años ha ido sustituyendo al correo ordinario, ya que además permite adjuntar archivos diversos.

HTTP: Acrónimo de HyperText Transfer Protocol. Protocolo que establece la comunicación entre un cliente y un servidor. Aparece al principio de todas las URL.

HTTPS: Conexión http segura y cifrada. Por ejemplo, cuando dejamos nuestros datos en un formulario de tarifas²⁴, siempre lo hacemos a través de una conexión segura.

World Wide Web: Sistema de información distribuido, basado en hipertexto, creado a principios de los años 90 por Tim Berners-Lee, investigador en el CERN, Suiza. La información puede estar en cualquier formato (texto, gráfico, audio, imagen fija o en movimiento) y es fácilmente accesible a los usuarios mediante los programas

navegadores. Es preciso destacar el hecho de que tanto Berners-Lee como el CERN renunciaron a la explotación comercial de este extraordinario invento.

JavaScript: Lenguaje de programación orientado a objetos, ejecutado en la máquina cliente (usuario), de licencia libre, desarrollado por Netscape, que interactúa con el código HTML utilizado para añadir interactividad a las paginas además de otras características.

Servidor: Software utilizado en el equipo host encargado de procesar los archivos a los navegadores para su visualización.

Oracle: Oracle es un sistema de gestión de base de datos objeto-relacional (o ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation.

Servlet: Los servlets son objetos que corren dentro y fuera del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad.

1.6 FORMULACIÓN DE HIPÓTESIS Y VARIABLES

1.6.1 Hipótesis General

Los estudiantes se atrasan en la elaboración de la tesis debido a sus múltiples ocupaciones diarias y se enfocan más en lo laboral atrasándose en el cronograma de actividades.

1.6.2 Hipótesis Específicas

La prolongación de tiempo al no entregar a su debido momento la tesis reduce el plazo de entrega del proyecto de tesis aumentando el costo del proceso a cada estudiante.

Se manipulan mucha información de los estudiantes al realizar el control y seguimiento de cada caso de forma manual demandando mucho tiempo.

El incremento de los estudiantes que cumplen con los requisitos para realizar el proceso de tesis y la cantidad de datos se incrementará aumentando la probabilidad de que ocurran errores y pérdida de información.

1.6.3 Matriz Causa-Efecto

PROBLEMA GENERAL	OBJETIVO GENERAL	HIPÓTESIS GENERAL
¿Qué factores inciden para que algunos estudiantes no cumplan a tiempo sus plazos para la elaboración de tesis?	Identificar los problemas que inciden para que algunos estudiantes no cumplan a tiempo sus plazos para la elaboración de tesis.	Los estudiantes se atrasan en la elaboración de la tesis por sus ocupaciones especialmente laborales por lo que descuidan y olvidan su cronograma de actividades de tesis.
SUB-PROBLEMAS	OBJETIVOS ESPECÍFICOS	HIPÓTESIS PARTICULARES
¿Qué consecuencia tiene que los estudiantes se atrasen al no entregar a tiempo la elaboración de la tesis?	Identificar las consecuencias que causaría a los estudiantes al no entregar a tiempo la elaboración de la tesis.	La prolongación de tiempo al no entregar a su debido momento la tesis reduce el plazo de entrega del proyecto de tesis aumentando el costo del proceso a cada estudiante.
¿En qué medida el proceso control y seguimiento de las tesis demanda tiempo?	Identificar en qué medida demanda tiempo el control y seguimiento de la tesis.	Se manipulan mucha información de los estudiantes al realizar el control y seguimiento de cada caso de forma manual demandando mucho tiempo.
¿Qué incidencias tiene el llevar a cabo el proceso manual y no automatizado?	Determinar las incidencias al llevar a cabo el proceso manual y no automatizado.	El incremento de los estudiantes que cumplen con los requisitos para realizar el proceso de tesis y la cantidad de datos se incrementará aumentando la probabilidad de que ocurran errores y pérdida de información.

Tabla 1 : Matriz Causa Efecto

Fuente: Los Autores

1.6.4 Variables

Variables Dependientes: Tiempo

Variables Independientes: Número de estudiantes con tesis atrasadas.

1.7 ASPECTOS METODOLÓGICOS DE LA INVESTIGACIÓN

1.7.1 Tipo de estudio

Investigación exploratorio: Esta investigación es exploratoria porque para tener solución a los problemas planteados se realizarán investigaciones en diferentes áreas en las cuales se obtengan procesos similares a los propuestos, permitiendo comprobar la efectividad abarcando todas las necesidades requeridas por el sujeto de estudio.

Investigación descriptiva: Es descriptiva porque describe todos los procesos y errores que son causados al momento de efectuar actividades en las cuales son incurridos gran cantidad de datos y la forma que afecta directa e indirectamente en el progreso de las actividades desarrolladas diariamente .

Investigación explicativa: Es explicativa porque se encarga de determinar cuáles son las causas que desde un principio han venido ocasionando problemas que deben tener soluciones inmediatas para mejorar el campo de trabajo.

Investigación de campo: Es de campo porque nos basamos en la observación de los acontecimientos y en los testimonios de las personas que laboran y realizan a diario estos procedimientos para de esta manera encontrar soluciones viables a la realidad.

Investigación no experimental: Es no experimental porque el objeto de estudio es observado en su contexto natural, en su realidad día a día sin realizarle cambios, analizándolos como se efectúan naturalmente en su ambiente en el que estos ocurren.

1.7.2 Método de investigación

Para llevar a cabo la ejecución de nuestro proyecto utilizaremos como método de investigación el método de análisis, el método experimental, el método comparativo los

cuales nos llevará a encontrar soluciones a los problemas planteados realizando una relación de causa-efecto.

1.7.3 Fuentes y técnicas para la recolección de información

Para la obtención de resultados se realizará una entrevista al Ing. Raúl Tingo Director de carrera de Sistemas en la Universidad Politécnica Salesiana, en lo que respecta la metodología de tesis para el control y seguimiento al Ing. Ricardo Naranjo.

Se realizará la recopilación de las bases teóricas y antecedentes para la investigación, recolectando, organizando y clasificando los datos obtenidos.

La Entrevista

Es una conversación dirigida con un propósito específico que utiliza un formato de preguntas y respuestas. En la entrevista se necesitan obtener las opiniones de los entrevistados.

La entrevista será dirigida al Ing. Raúl Tingo Director de la carrera de Ingeniería de Sistemas en la Universidad Politécnica Salesiana sede Guayaquil, en la cual se efectuará una conversación para llegar a un concepto específico de cuáles son los procesos que actualmente realiza manualmente y de que mejoras habría al implementar la automatización de dichos procesos.

Encuesta

Se realizará una encuesta a los estudiantes egresados para determinar cuáles son los factores en si inciden al momento de realizar la tesis para que ellos determinen si las actividades se están realizando en un tiempo óptimo y manifiesten su aceptación o descontento de los procesos que se realizan en el departamento que a la vez estos influyen en los procedimientos de la institución.

1.7.4 Población y Muestra

“Desde el punto de vista estadístico, una población o universo de estudio puede estar referido a cualquier conjunto de sus elementos de los cuales se pretende indagar y conocer sus características, o una de ellas, y para el cual serán válidas las conclusiones obtenidas en la investigación”

“La muestra estadística es una parte de la población, o sea, un número de individuos u objetos seleccionados científicamente, cada uno de los cuales es un elemento del universo. La muestra es obtenida con el fin de investigar, a partir del conocimiento de sus características particulares, las propiedades de una población”

La población seleccionada para el análisis exploratorio ha sido los estudiantes pertenecientes a la Universidad Politécnica Salesiana de la carrera de Ingeniería de Sistemas que estén egresados, con tema de tesis aprobado y que estén en la actualidad realizando el proceso de tesis que nos ayudarán a obtener información y los correspondientes datos estadísticos del objeto de estudio.

1.8 Resultados Esperados

Lo que se espera al implementar el proyecto es llevar el procesos de tesis de una manera automatizada, donde se pueda realizar vía web(desde cualquier computadora conectada a internet), para que así los estudiantes no tengan interrupción en el trabajo o tengan que acercarse a la institución a realizar dicho procedimiento.

Se espera también, tener un mayor seguimiento entre estudiantes, tutores y administrativo, pues al ya no utilizarse papel para este proceso se evitaría el tras papeleo o perdida de algún documento, lo cual permitirá tener resultados de manera inmediata y más precisa.

CAPITULO 2

2. ANÁLISIS PRESENTACIÓN DE RESULTADOS Y DIAGNÓSTICO

2.1 Organigrama de Gestión de Temas de Tesis

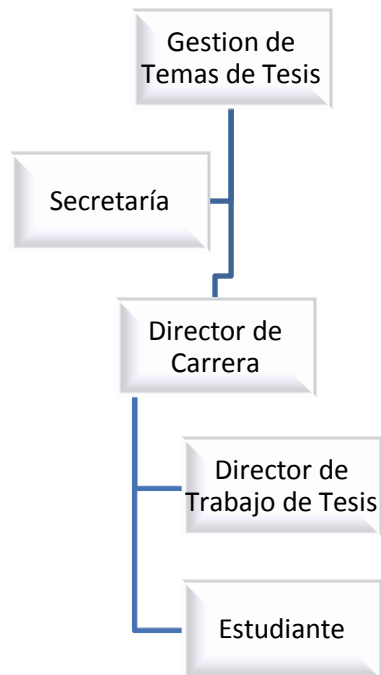


Ilustración 5: Organigrama de la Gestión de Temas de Tesis

Fuente: Los Autores

Descripción de las principales funciones:

Director de Carrera: Es la persona encargada junto a miembros del consejo de carrera de Sistemas aceptar los temas de tesis emitiendo la resolución con fecha de inicio y fecha fin de entrega de tesis, ejerciendo en la actualidad Ing. Raúl Tingo.

Director de Trabajo de Tesis: Está a cargo de asistir a los requerimientos de los estudiantes con el tema de tesis brindándoles ayuda y conocimientos para terminar la tesis.

Secretaria: Es la persona encargada de realizar la resolución del tema de tesis aprobado ayudando a los alumnos con la debida información del trámite y los papeles a presentar.

Estudiantes: Sustenta el tema de Tesis con las reglas establecidas.

2.2 Herramientas de Análisis

2.2.1 Análisis de los Procesos actuales

2.2.1.1 Proceso De La Denuncia de Tesis.

Una vez que el estudiante ha cumplido al menos el 80% del total de materias correspondientes a cada titulación, solicita en la Secretaría de Campus el certificado habilitante que le permite matricularse en denuncia de trabajo de grado o tesis.

El estudiante se matricula en denuncia de Tesis en Tesorería y presenta por escrito en Secretaría de Campus la denuncia del tema y del plan de Trabajo de Tesis, con la recomendación de un profesor de la Facultad, dirigida al Director de Carrera y en los formatos establecidos por cada Facultad.

Secretaría de Campus entregará al estudiante la constancia de la entrega-recepción y envía, dentro de las siguientes 8 horas laborables, la solicitud al Director de Carrera, previo ingreso en los registros respectivos.

El Director de Carrera deberá presentar la solicitud de denuncia del tema y plan en el próximo Consejo de Carrera.

El Consejo de Carrera designa al profesor Director del Trabajo de Tesis. Esta designación será comunicada por escrito a través de la Secretaría.

El designado Director de Trabajo de Tesis deberá entregar su informe en el siguiente Consejo de Carrera ordinario, a través del Director de Carrera, para su aprobación. En este mismo Consejo se ratificaría al Director de Trabajo de Tesis.

Si el informe no es favorable, el estudiante podrá presentar nuevamente el Plan, con las recomendaciones sugeridas. Esta resolución deberá ser comunicada a través de Secretaría en un plazo de 8 horas laborables.

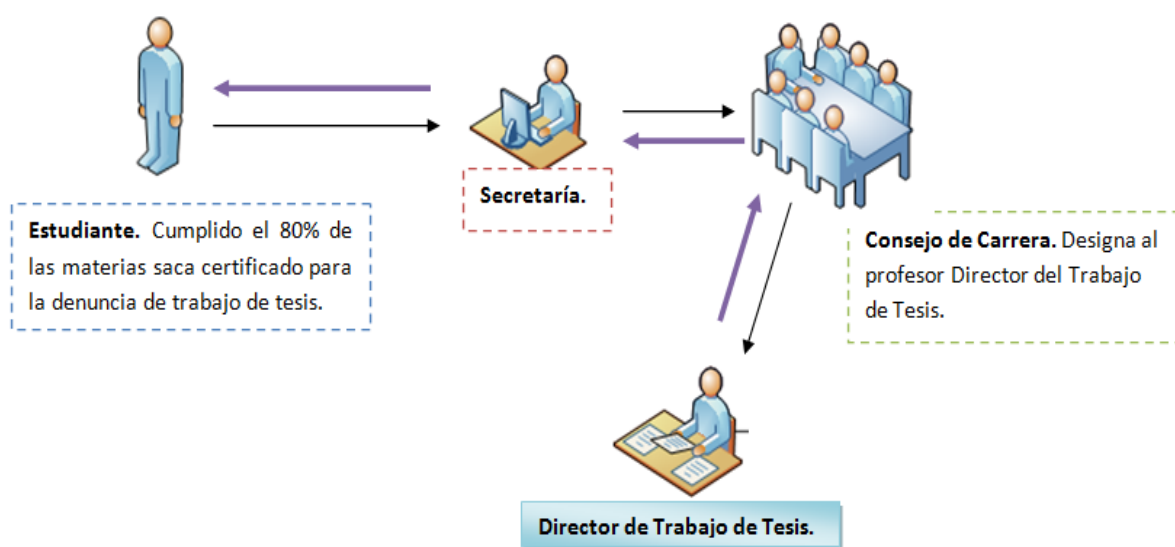


Ilustración 6: Proceso de Denuncia de Tesis

Fuente: Los Autores

2.2.1.2 Proceso De Desarrollo De La Tesis.

El estudiante, una vez aprobado el Plan de Trabajo de Tesis, iniciará el desarrollo del mismo, con la asesoría del Director de Trabajo de Tesis. Esta resolución será notificada al estudiante a través de Secretaría. El Director de Trabajo de Tesis informará por escrito mensualmente al Director de Carrera, los avances del Trabajo de Tesis.

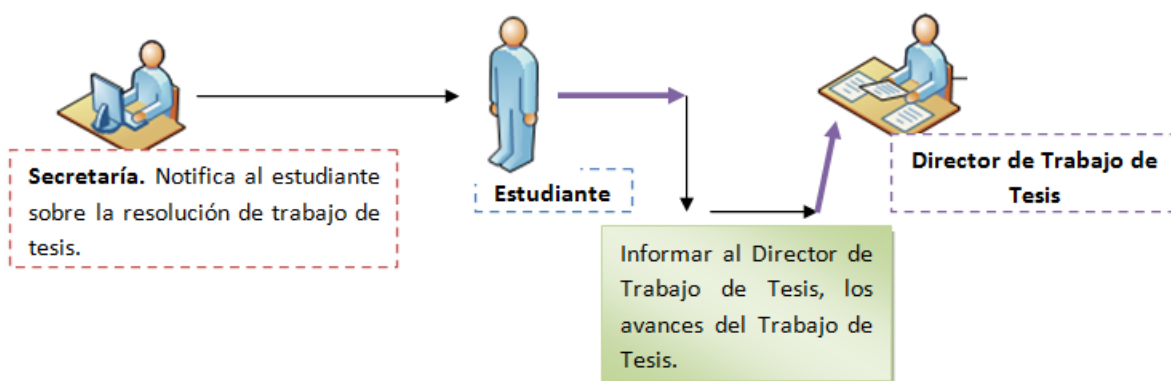


Ilustración 7: Proceso De Desarrollo de la Tesis
Fuente: Los Autores

2.2.1.3 Proceso De Petición de Prorroga.

Si el estudiante aún no ha concluido en el plazo establecido de 6 meses su trabajo de tesis procederá a la petición de una prórroga deberá acercarse a tesorería cancelar dicho valor.

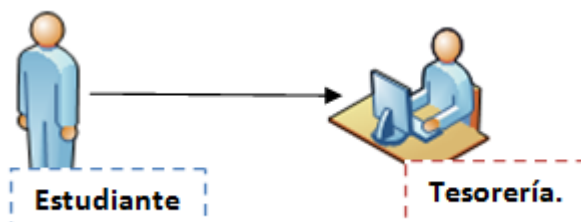


Ilustración 8: Proceso de Petición de Prorroga
Fuente: Los Autores

2.2.1.4 Proceso De La Presentación Del Trabajo Escrito.

Una vez concluido el Trabajo de Tesis, el Director del mismo entregará en Secretaria de Campus un informe dirigido al Director de Carrera, el que incluye la calificación del trabajo escrito sobre 100 puntos presentado en el formato correspondiente La Secretaria de Campus adjuntará al informe, la siguiente documentación presentada por el estudiante:

- a. Tres ejemplares del Trabajo de Grado o Tesis empastados,
- b. Un respaldo en formato digital (CD),
- c. El abstract (resumen o síntesis) con una extensión no mayor a 3 páginas,
- d. Comprobantes de pago de los derechos de grado.

Se deberá llenar el acta de entrega-recepción, entre el estudiante y la Secretaría de Campus.

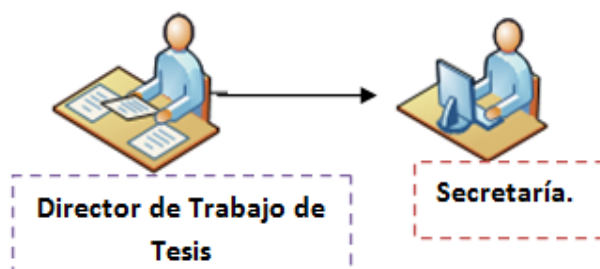


Ilustración 9: Proceso De La Presentación Del Trabajo Escrito

Fuente: Los Autores

2.2.1.5 Proceso De Declaratoria De Aptitud Para Sustentación.

Una vez receptado dicho informe, secretaria de campus remitirá en un plazo de diez días laborables, al Director de Carrera toda la documentación requerida para la declaración de aptitud para la sustentación:

- a. Certificado de matrícula de cada uno de los períodos de estudio de la carrera.
- b. Certificado de aprobación de los créditos por materias, y demás requisitos para académicos vigentes en el pensum de estudios cursado.
- c. Informe del Director de Trabajo Tesis con la calificación total del trabajo escrito de Tesis.

El Director de Carrera, deberá elaborar un informe que será presentado en el siguiente Consejo de Carrera, respecto del cumplimiento de los requisitos legales y reglamentarios para la sustentación.

El Consejo de Carrera, previo informe favorable del Director de Carrera, declarará la aptitud del estudiante y determinará la fecha de sustentación para optar por la titulación respectiva, y entregará a los miembros del tribunal de sustentación, a través de secretaria, un ejemplar empastado del trabajo escrito, el cual se devolverá el día de la sustentación.

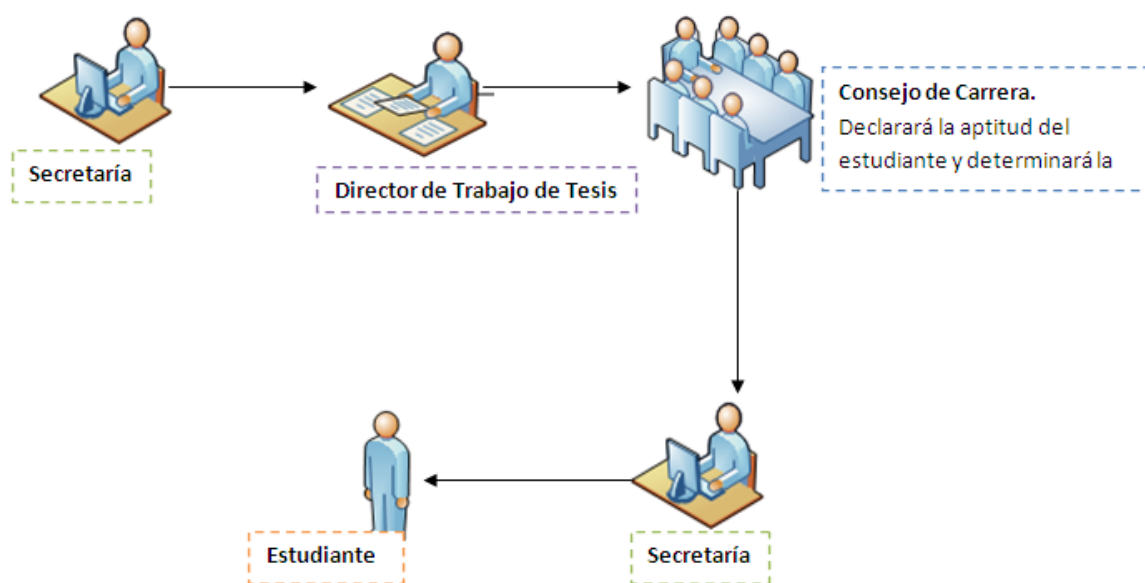


Ilustración 10: Proceso De Declaratoria De Aptitud Para Sustentación

Fuente: Los Autores

2.2.1.6 Proceso De La Sustentación.

El Tribunal conformado por el Director de Carrera quien preside, el Director de Trabajo de Tesis y un docente designado por el Consejo de Carrera, calificará la sustentación del Trabajo de Tesis sobre cien puntos. La nota de sustentación final que el estudiante obtenga será el promedio de las tres notas. Si el estudiante no obtuviere como mínimo setenta puntos sobre cien en la nota de sustentación, se presentará a una nueva sustentación en un plazo no menor a treinta días laborables, y de no aprobar en esta oportunidad el Trabajo de Tesis, se anulará y podrá denunciar otro tema por una sola vez.

En caso de ser más de un estudiante, la nota deberá ser individual.

Terminada la sustentación, Secretaria distribuirá los ejemplares empastados de la siguiente manera: Uno al Director del Trabajo de Tesis, otro para la biblioteca, y el restante para el o los estudiantes.

2.2.1.7 Proceso De La Declaratoria De Aptitud Para Titulación

Una vez cumplido con todos los requisitos legales, el Director de Carrera solicitará al Decano continuar con las gestiones pertinentes para conceder la respectiva titulación.

El Decano solicitará al Consejo Directivo de Facultad la declaratoria de aptitud para la Titulación respectiva. Con dicha resolución tramitará ante la Secretaría General la emisión de la especie del título. Las menciones se tramitaran ante la Secretaria de Campus.

El Vicerrector de la Sede, en coordinación con los Directores de Carrera, determinará la fecha de investidura, la cual será comunicada al Decano para los fines pertinentes.

NOTA: En la investigación se ha tomado en consideración el punto 2.2.1.2. ya que los siguientes dependen de otras áreas.

2.3 Análisis estadístico de las encuestas

Este estudio fue realizado en la ciudad de Guayaquil en la Universidad Politécnica Salesiana dirigida a estudiantes egresados de la carrera de Ingeniería de Sistemas, por diferentes medios vía telefónica, web y escrita.

El análisis fue realizado después de haber implementado el sistema. Obteniendo los resultados de las encuestas realizadas se demostrará la efectividad del uso del sistema.

1. ¿Usted Es?

		%
Estudiante	0	0
Egresado	35	88%
Graduado	5	13%
Total	40	100%

Tabla 2: Resultado de la Encuesta Pregunta 1

Fuente: Los Autores

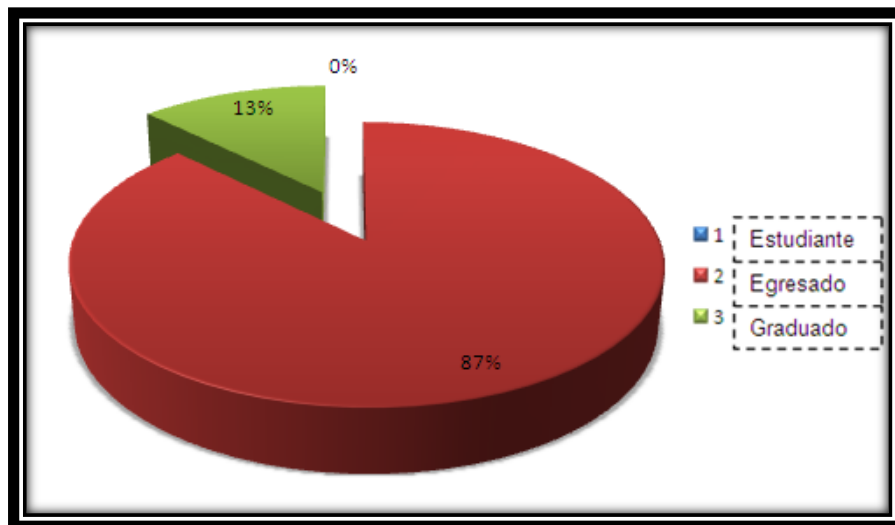


Ilustración 11: Gráfico Estadístico resultados de la encuesta pregunta 1

Fuente: Los Autores

De las encuestas realizadas a 40 estudiantes de la UPS. Se determinó que el 88% estaban egresados y el 5% ya se habían incorporado.

2. ¿Usted Solicitó Prorroga Para culminar el Trabajo de Tesis?

		%
SI	30	75%
NO	10	25%
Total	40	100

Tabla 3: Resultado de la Encuesta Pregunta 2

Fuente: Los Autores

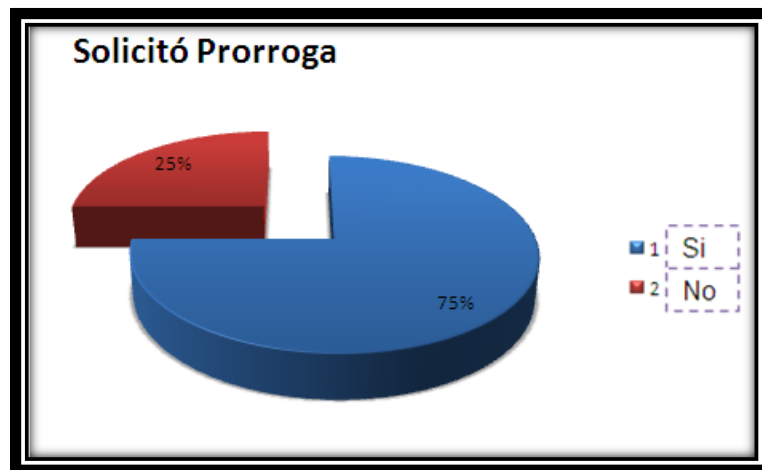


Ilustración 12: Gráfico Estadístico resultados de la encuesta pregunta 2

Fuente: Los Autores

El 75% de los estudiantes manifestaron que si solicitaron prorroga mientras que el 25% no la solicitaron.

3. ¿Cuál Fue Su Principal Problema por el que se atrasaron al Realizar La Tesis?
(Respuestas Múltiple).

		%
Laboral	17	32%
No tiene presente la fecha de entrega de tesis	10	19%
No cumplió a tiempo con el cronograma	10	19%
Falta de conocimientos	7	13%
Económicos	3	6%
Otro.	3	6%
Problemas Familiares	2	4%
Salud	1	2%
Total	53	100%

Tabla 4: Resultado de la Encuesta Pregunta 3
Fuente: Los Autores

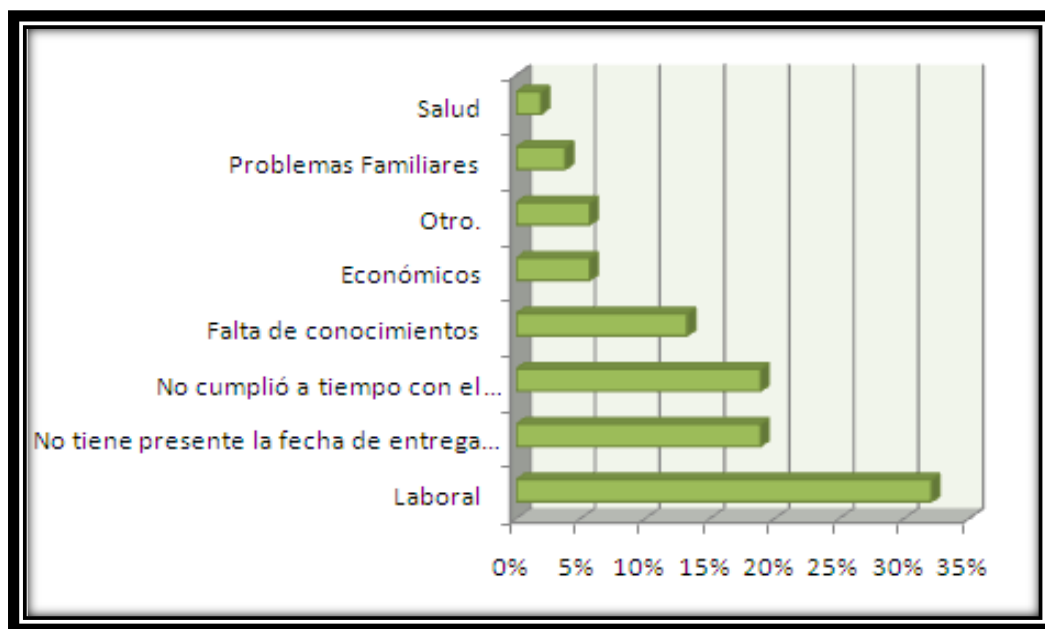


Ilustración 13: Gráfico Estadístico resultados de la encuesta pregunta 3

Fuente: Los Autores

Se determinó que el 32% de los estudiantes no terminan a tiempo su trabajo de tesis por motivos laborales, 19% por qué no cumplieron a tiempo con el cronograma, 19% porque no tienen presente la fecha de entrega.

4. ¿Cree Usted Que Sería Factible Enviarles Recordatorios Al Email Sobre El Tiempo Límite Para Realizar La Tesis?

		%
SI	36	90%
NO	4	10%
Total	40	100%

Tabla 5: Resultado de la Encuesta Pregunta 4

Fuente: Los Autores

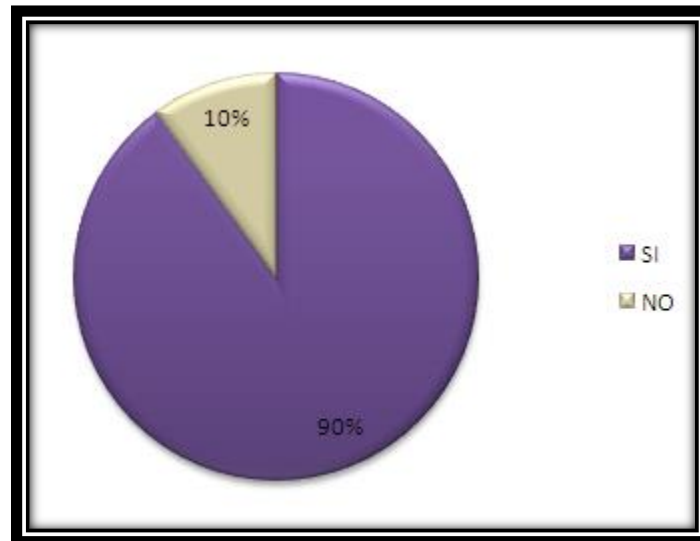


Ilustración 14: Gráfico Estadístico resultados de la encuesta pregunta 4

Fuente: Los Autores

Se determinó que el 90% de los estudiantes desean que les envíen recordatorios sobre la fecha límite para realizar la tesis y el 10% no requieren.

5. ¿Desearía un mayor control y monitoreo por parte del tutor sobre la tesis?

		%
SI	39	97%
NO	1	3%
Total	40	100%

Tabla 6: Resultado de la Encuesta Pregunta 5

Fuente: Los Autores

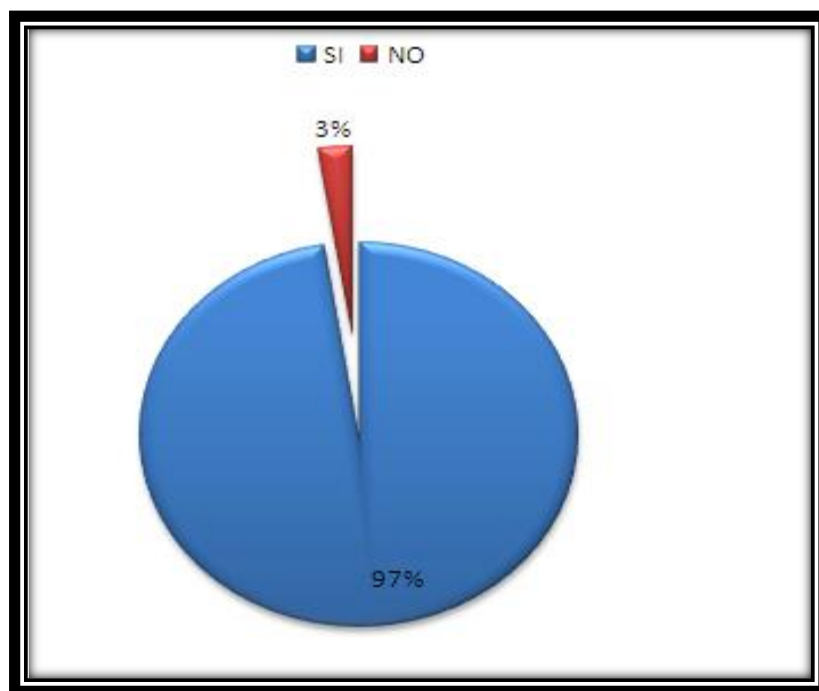


Ilustración 15: Gráfico Estadístico resultados de la encuesta pregunta 5

Fuente: Los Autores

Se determinó que el 97% de los estudiantes requieren un mayor control de la tesis y un 3% de que no requieren con una totalidad del 100% de los 40 estudiantes que se le realizó las encuestas.

CONCLUSIONES DEL TRABAJO DE CAMPO

De acuerdo a los resultados de las encuestas realizadas a 40 estudiantes de la Universidad Politécnica Salesiana se ha determinado:

Existe un gran porcentaje de estudiantes egresados y graduados que han solicitado prórroga por motivo laboral del 32%, un 19% por qué no cumplieron a tiempo con el cronograma y porque no tienen presente la fecha de entrega su trabajo de tesis por lo que requieren un mayor control sobre el trabajo de grado tesis y tiempo de entrega.

CAPITULO 3

ANÁLISIS Y DISEÑO DEL SISTEMA PROPUESTO.

3.1 ARQUITECTURA DEL SISTEMA

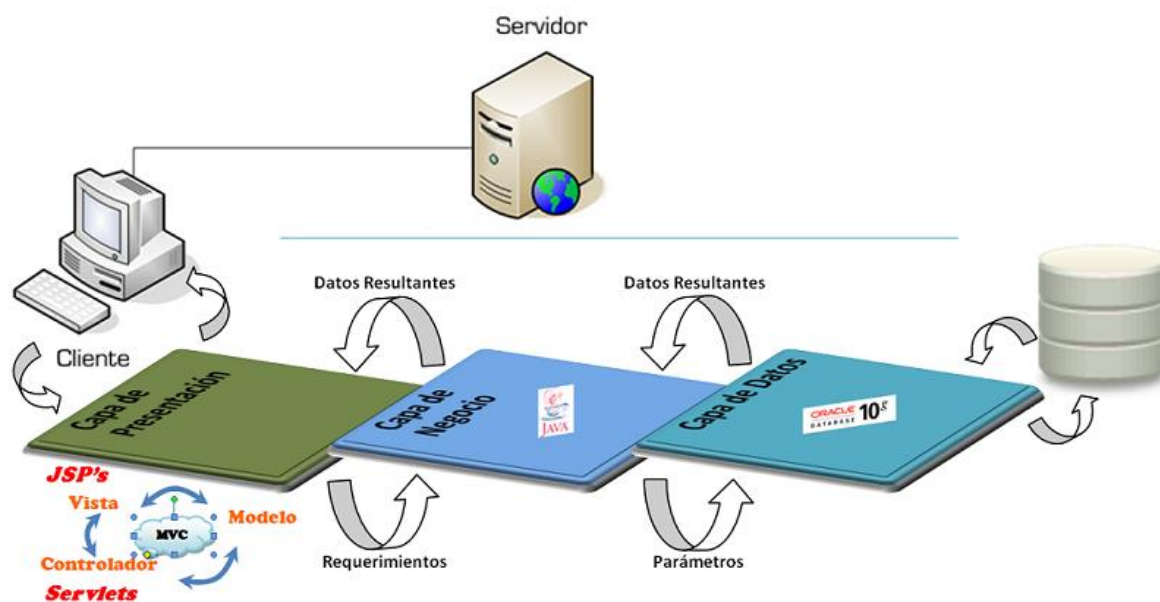


Ilustración 16: Arquitecta MBC

Fuente: Los Autores

3.2 MODELOS DE ANÁLISIS

3.2.1 Enfoque de Procesos

En esta sección, explicaremos los esquemas que hemos utilizado para representar la Automatización del Proceso de Control y Monitoreo de Elaboración de Tesis.



Ilustración 17: Modelo de Procesos

Fuente: Los Autores

Se detallarán cada uno de los procesos a automatizarse, las actividades y tareas que se involucran cada uno de ellos.

Es importante señalar que los procesos y actividades identificados en el gráfico, son los que nuestra aplicación automatizará.

PROCESO CLAVE: Monitoreo de Tesis.

Es el proceso de control y monitoreo de tesis que se realiza en la Gestión de temas de Tesis tales como: el ingreso de estudiantes que solicitan prorroga, el ingreso y desactivación de tutores, consulta de reportes, recordatorios, lo cual involucra todas las actividades que deben realizarse desde el momento en que el estudiante egresa o cumple con el 80% de las materias.

ACTIVIDADES

Crear Usuarios: Esta actividad incluye la creación y la desactivación de tutores para el ingreso al sistema web.

Crear Monitoreo: Esta actividad incluye el registro y consulta del monitoreo de Tesis por parte de los Tutores para así facilitar un mayor control del cronograma de actividades que realiza el estudiante al presentar su propuesta de tesis.

Crear Prorrogas: Esta actividad incluye el registro de los estudiantes que no culminan su trabajo de tesis en el plazo establecido y solicitan prórroga para la extensión de tiempo para la presentación de su trabajo de tesis.

Reporte: Consulta de Tesis activas, Control y seguimiento de Tesis, estudiantes que han solicitado prorrogas y reporte de cuantas tesis un tutor tiene a su cargo.

Envío de Email: Recordatorios vía email a los estudiantes del plazo que tienen en entregar su tema de tesis y el porcentaje de avance de su trabajo de tesis.

3.2.2 DIAGRAMA DE PROCESOS DEL SISTEMA

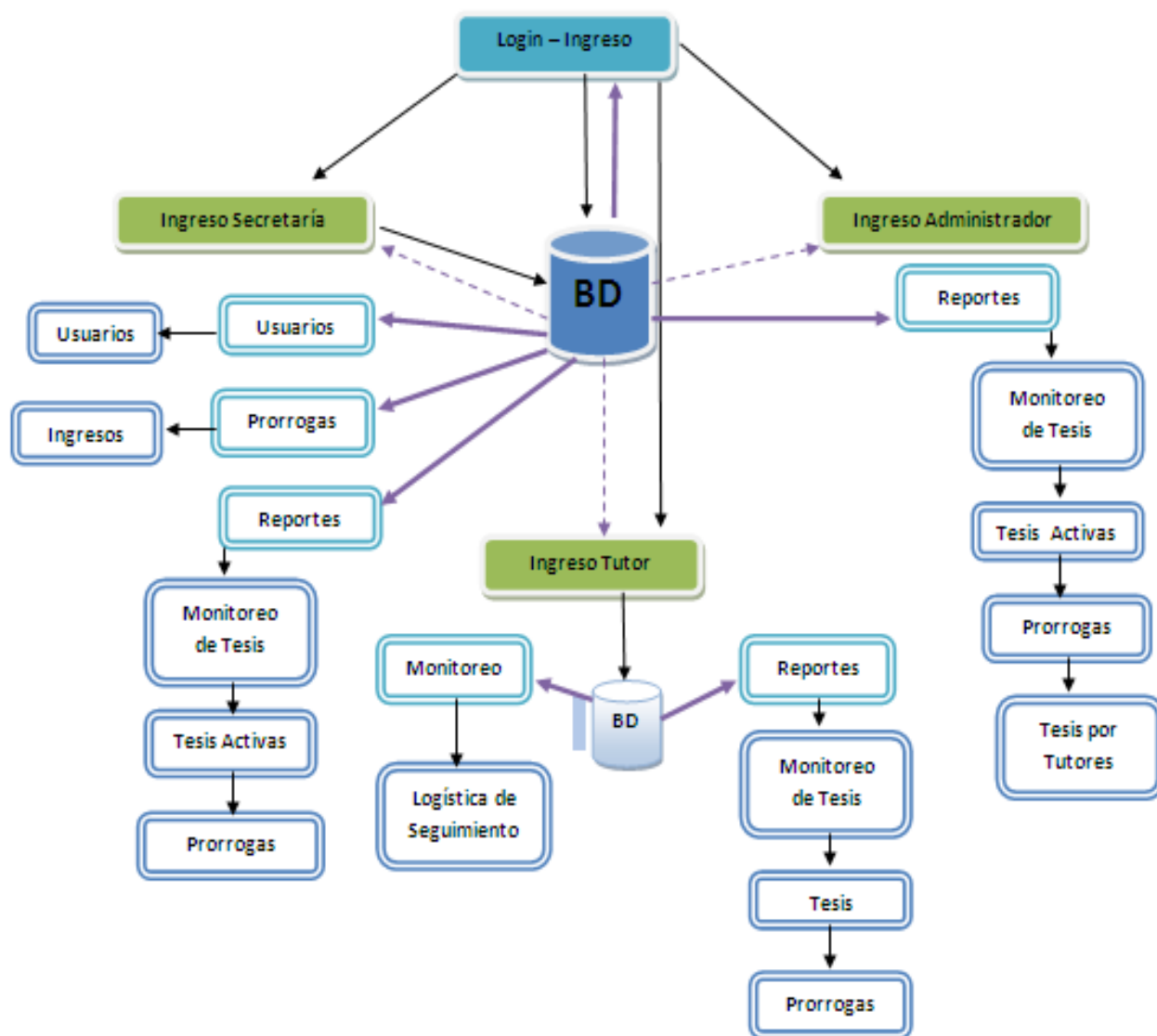


Ilustración 18: Procesos del Sistema

Fuente: Los Autores

MODULO DE USUARIOS

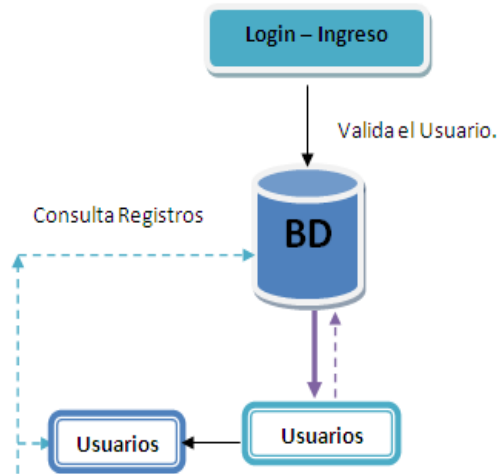


Ilustración 19: Proceso Usuarios

Fuente: Los Autores

MODULO DE PRORROGAS

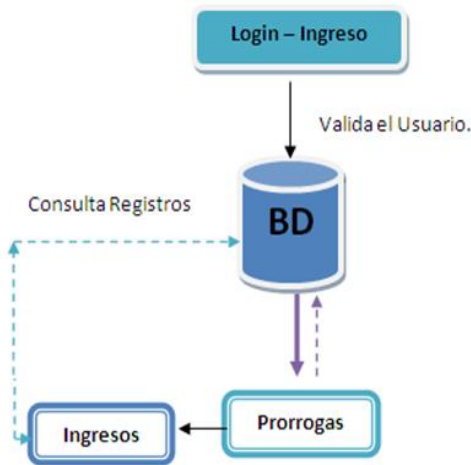


Ilustración 20: Proceso Prorrogas

Fuente: Los Autores

MODULO DE MONITOREO

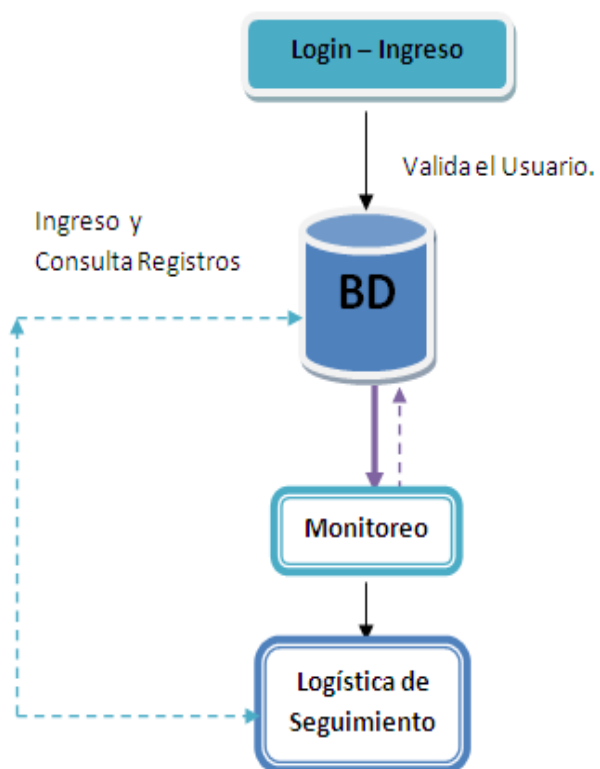


Ilustración 21: Proceso Monitoreo

Fuente: Los Autores

MODULO DE REPORTE

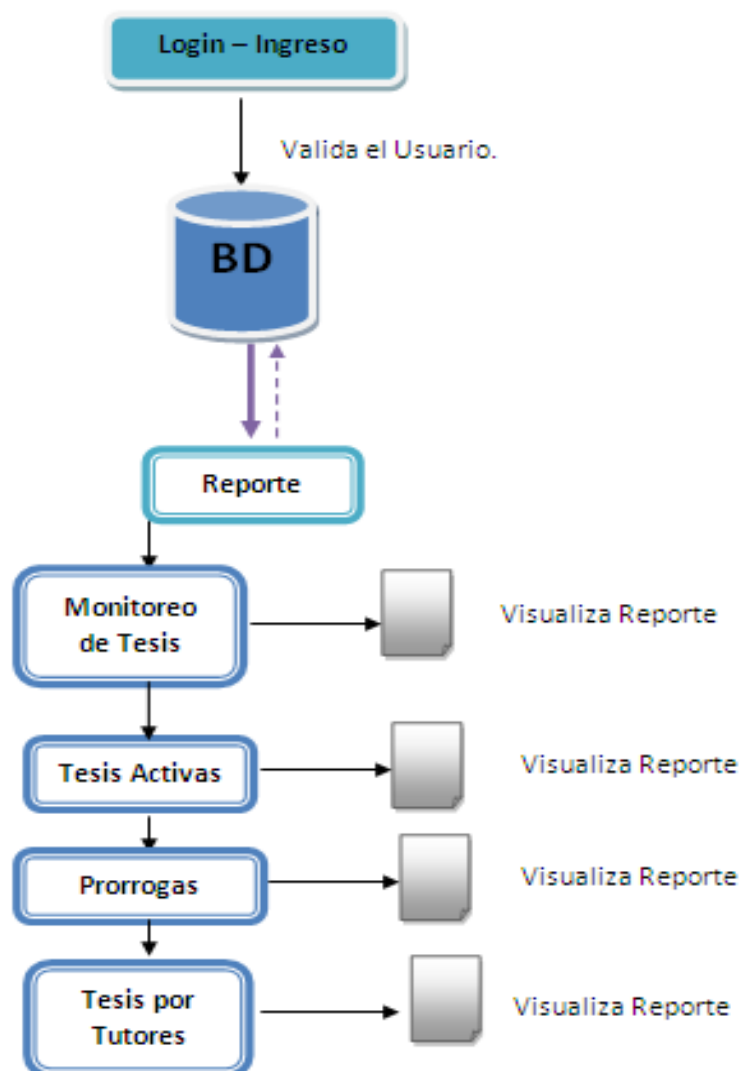


Ilustración 22: Proceso Reporte

Fuente: Los Autores

3.2.3 DIAGRAMAS DE FLUJO DE DATOS

MÓDULO USUARIOS

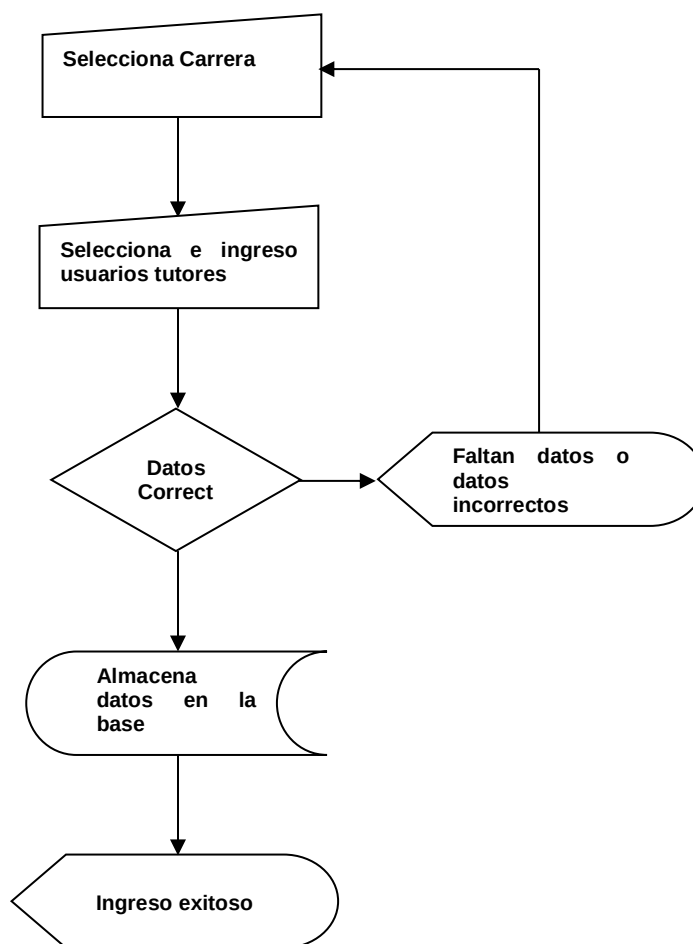


Ilustración 23: Diagrama de Flujo del Módulo de Usuarios

Fuente: Los Autores

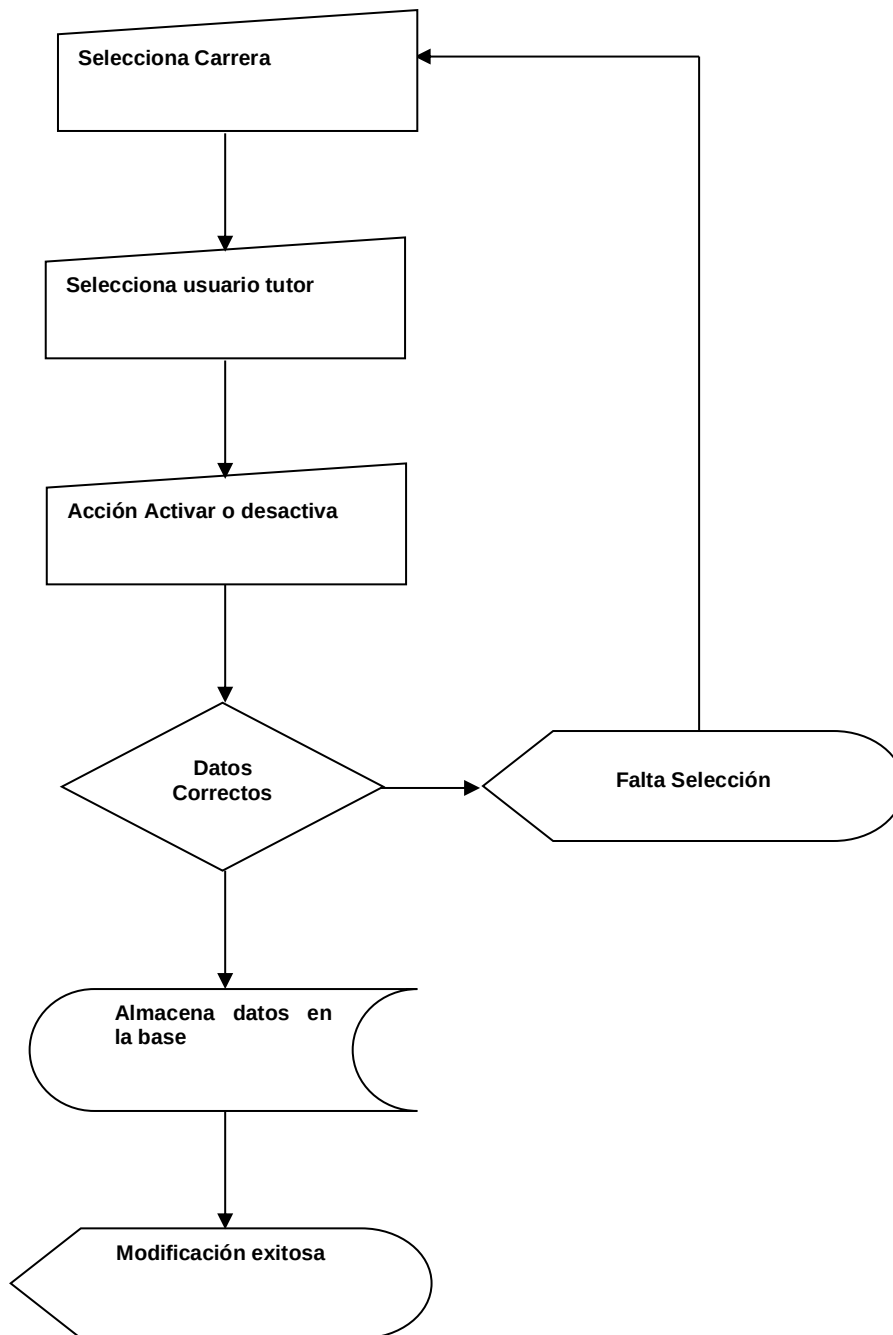


Ilustración 24: Diagrama de Flujo del Módulo de Usuarios

Fuente: Los Autores

MODULO DE PRORROGAS

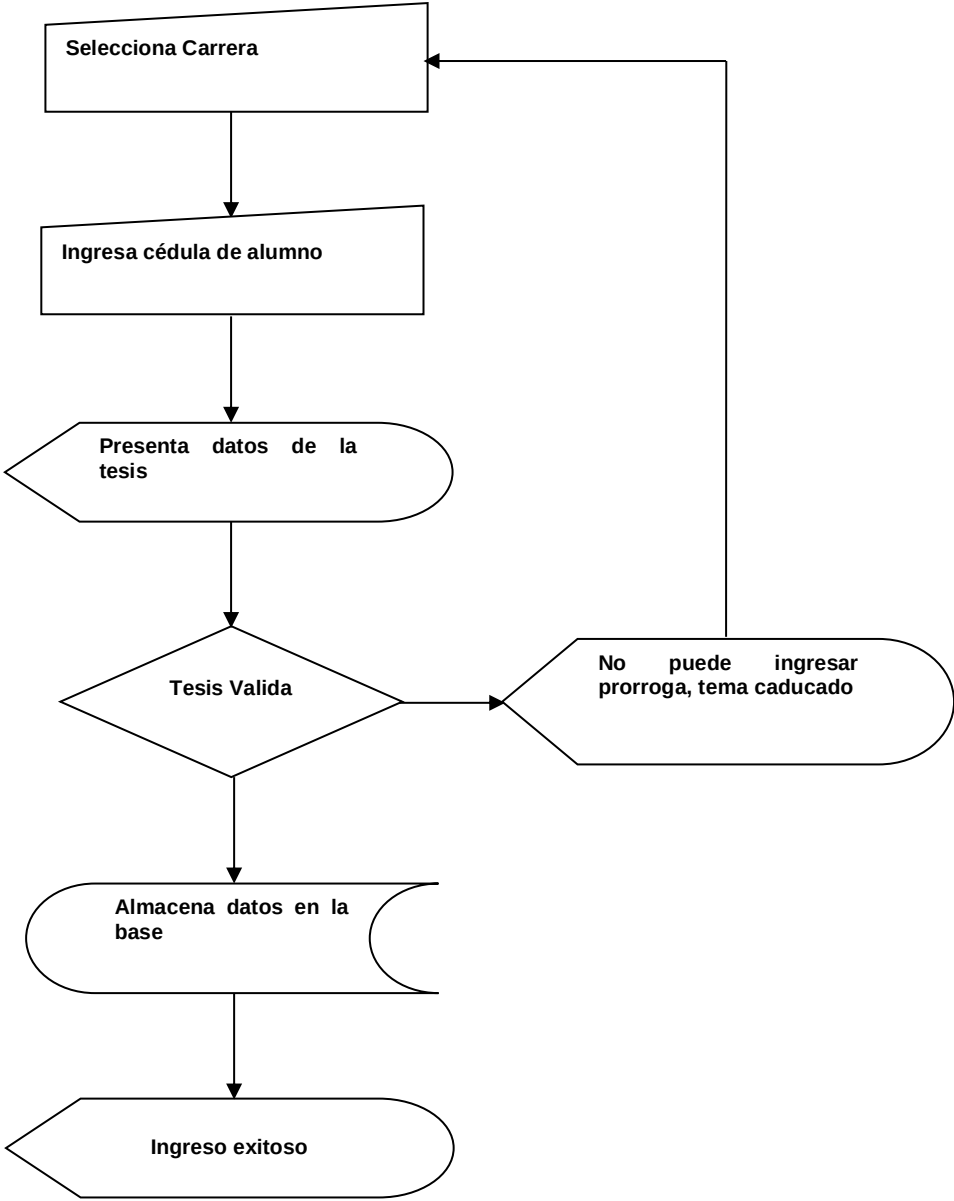


Ilustración 25: Diagrama de Flujo del Módulo de Prorrogas

Fuente: Los Autores

MODULO DE MONITOREO

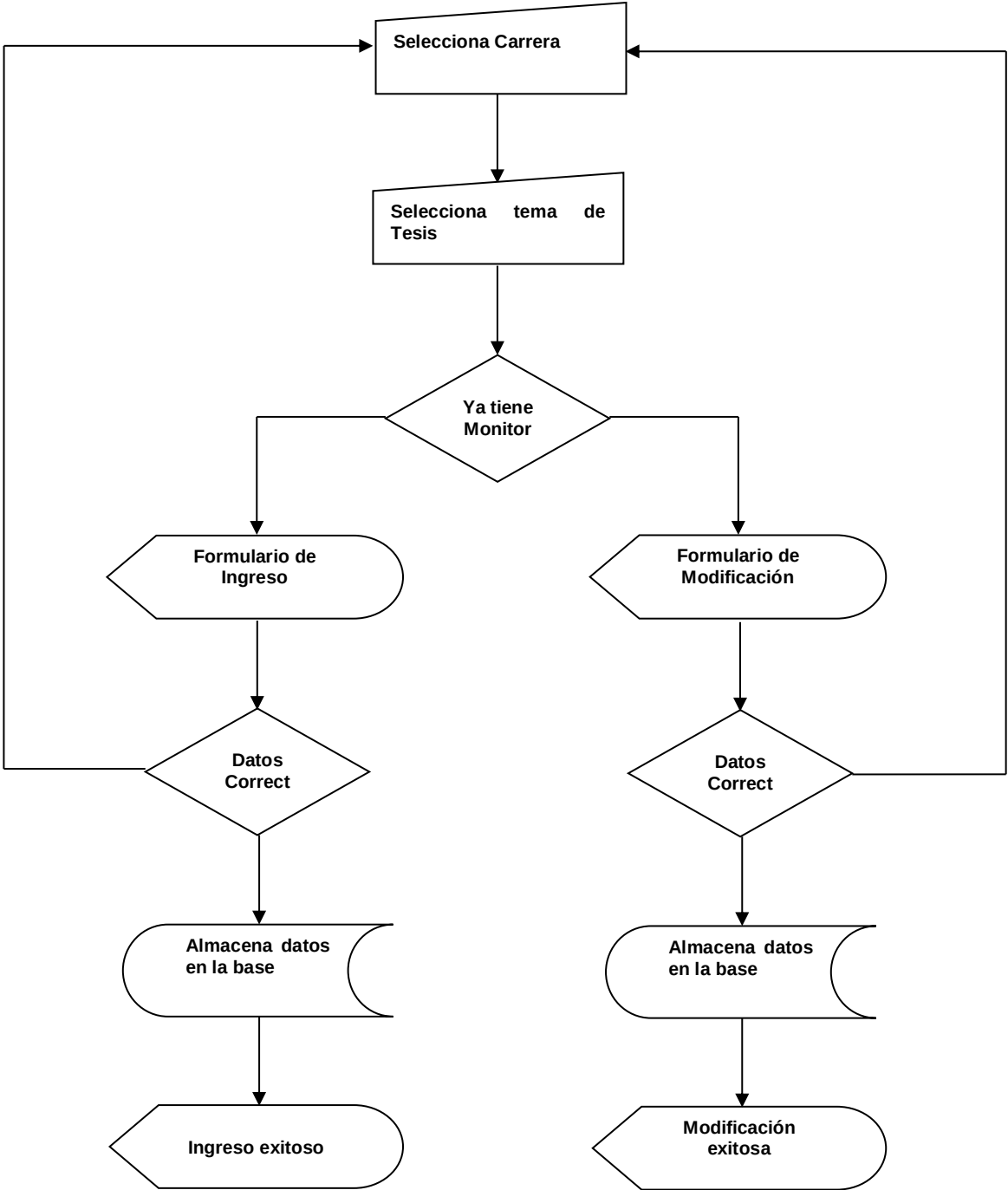


Ilustración 26: Diagrama de Flujo del Módulo de Monitoreo

Fuente: Los Autores

3.2.4 DIAGRAMA DE CASOS DE USO

Se han determinado los siguientes actores.

1. DIRECTOR DE CARRERA.

Tiene la facultad de acceder a toda la información referente a los trabajos de grado como por ejemplo las solicitudes de prórrogas por el estudiante, el control y monitoreo del trabajo de grado y los temas de tesis que cada tutor se le ha asignado.

2. Tutores.

Realiza el registro y control del seguimiento de todos los trabajos de grado que tiene a su cargo, también podrá visualizar los reportes de las tesis que ofrece tutoría.

3. Secretaría.

Realiza el control de Registro y eliminación de Tutores. El ingreso de Prórrogas que el alumno solicita previa validación embebida dentro del sistema.

Se han definido los siguientes casos de uso:

MÓDULO DE USUARIOS

1. Ingresar usuario: Realiza el ingreso de usuario y clave para el acceso de los tutores en el sistema.

2. Desactivar Tutores: Desactiva a los tutores.

MÓDULO DE PRORROGAS

1. Ingreso de Cedula del Estudiante: Realiza una consulta de los estudiantes para el ingreso de la Prorroga.

MÓDULO DE MONITOREO

1. Selección e Ingreso de Temas de Tesis: Selecciona el tema al que se le va hacer el seguimiento y control de la tesis.

2. Consulta: Realiza la consulta del control de tema de tesis.

Caso de Uso Autenticación del Usuario

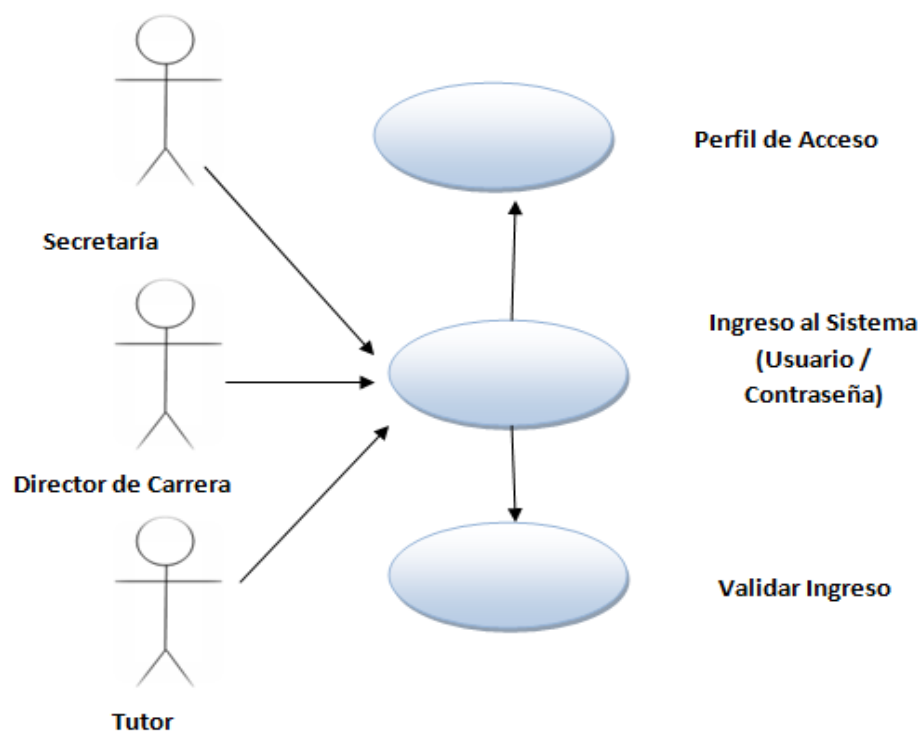


Ilustración 27: Autenticación de usuario

Fuente: Los Autores

NOMBRE:	Usuarios
AUTOR:	Secretaría , Base de Datos
FECHA:	15/09/2012
OBJETIVO	Ingreso al sistema
DESCRIPCION:	Para poder ingresar al sistema es necesario obtener un usuario y clave de acceso.
PRECONDICIONES:	Ninguna
SECUENCIAS:	El usuario digita su usuario y clave de acceso. El sistema comprueba la información. El sistema permite acceso según el usuario.

Tabla 7: Descripción de Autenticación de usuario

Fuente: Los Autores

Caso de Uso de Usuarios

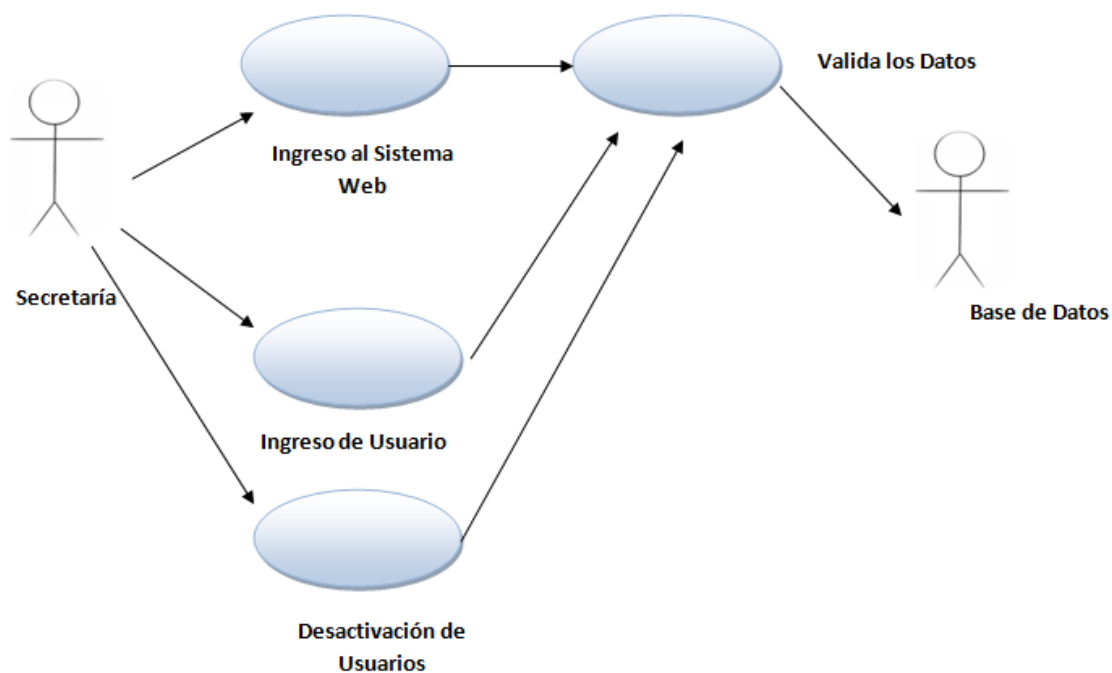


Ilustración 28 : Creación de Usuarios

Fuente: Los Autores

NOMBRE:	Autenticación
AUTOR:	Secretaría , Director, Tutor
FECHA:	15/09/2012
OBJETIVO	Ingreso de Usuarios
DESCRIPCION:	se registran a los tutores para que puedan acceder a la página web
PRECONDICIONES:	Ninguna
SECUENCIAS:	Deberá seleccionar el tutor Activarlo o desactivarlo según sea el caso.

Tabla 8: Descripción de Creación de Usuarios

Fuente: Los Autores

Caso de Uso Prorrogas

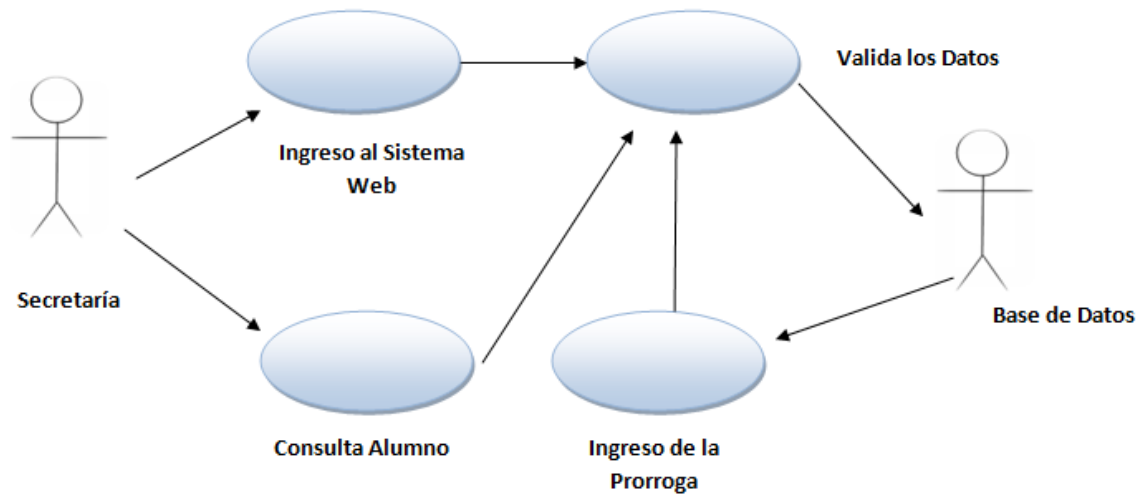


Ilustración 29: Creación de Prorroga

Fuente: Los Autores

DESCRIPCION CASO DE USO: Prorroga	
ACTORES	Secretaría, Base de datos.
OBJETIVO	Ingresar al sistema web como Secretaría a la opción de Prorrogas.
DESCRIPCION	Secretaría consultará e ingresará la prorroga
PRECONDICION	Haber ingresado como usuario secretaria.
SECUENCIA	El usuario deberá consultar el alumno por el número de cédula para luego ingresar la prórroga.

Tabla 9: Descripción de Creación de Prorroga

Fuente: Los Autores

Caso de Uso Monitoreo

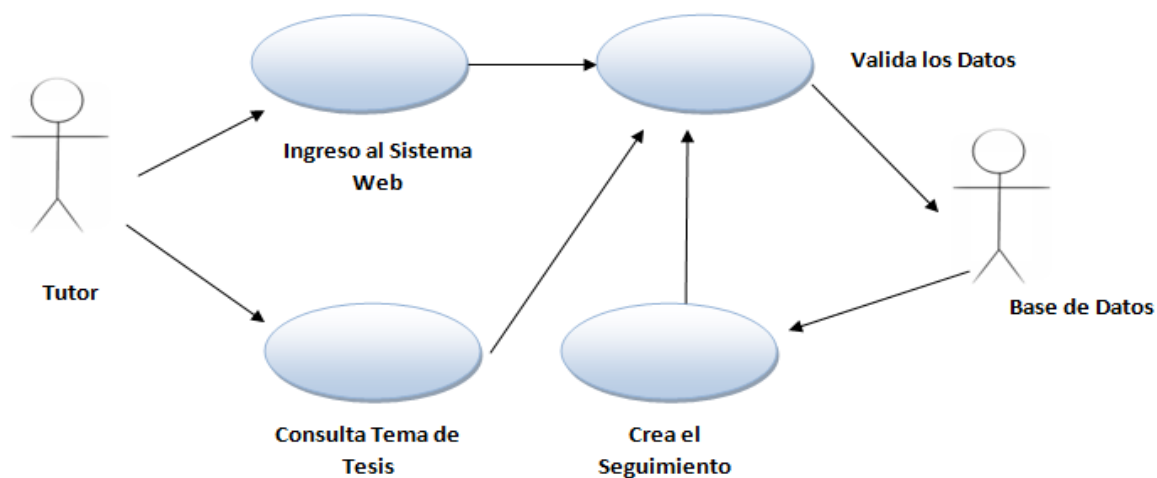


Ilustración 30: Creación de Monitoreo

Fuente: Los Autores

DESCRIPCION CASO DE USO: Monitoreo	
ACTORES	Tutores, Base de datos.
OBJETIVO	Ingresar y Consultar el Seguimiento de Tesis.
DESCRIPCION	Los tutores podrán crear y consultar el seguimiento de la tesis.
PRECONDICION	Haber ingresado como usuario Tutores y consultar por el Tema de Tesis.
SECUENCIA	El tutor debe seleccionar la carrera y la tesis a crear o consultar. Debe seleccionar los subcapítulos que el estudiante ha realizado.

Tabla 10: Descripción de Creación de Monitoreo

Fuente: Los Autores

Caso de Uso Reportes

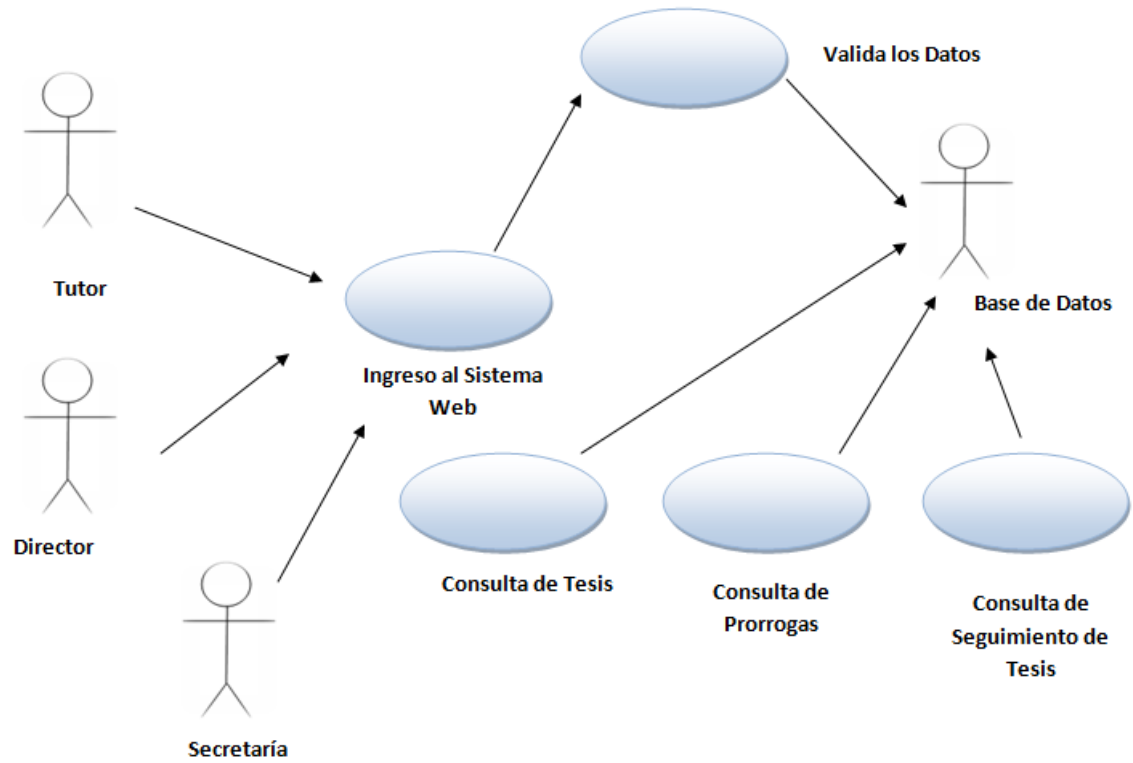


Ilustración 31: Creación de Reportes

Fuente: Los Autores

DESCRIPCION CASO DE USO: Reportes	
ACTORES	Tutores, Secretaría, Director, Base de datos.
OBJETIVO	Visualización de Consultas
DESCRIPCION	Se podrá visualizar las consultas de tesis activas , alumnos que han solicitado prorrogas, el control y seguimiento de tesis
PRECONDICION	Haber ingresado como usuario Tutores y consultar por el Tema de Tesis.
SECUENCIA	Al escoger la opción podrá observar un listado de tesis activas o de alumnos que han solicitado prorroga.

Tabla 11: Descripción Creación de Monitoreo

Fuente: Los Autores

3.2.5 DIAGRAMAS DE ESTADOS

Los presentes diagramas representan todos los estados que pueden tener cada uno de los casos de usos más relevantes que se han seleccionado, para entender de una mejor manera el comportamiento de cada uno de ellos.

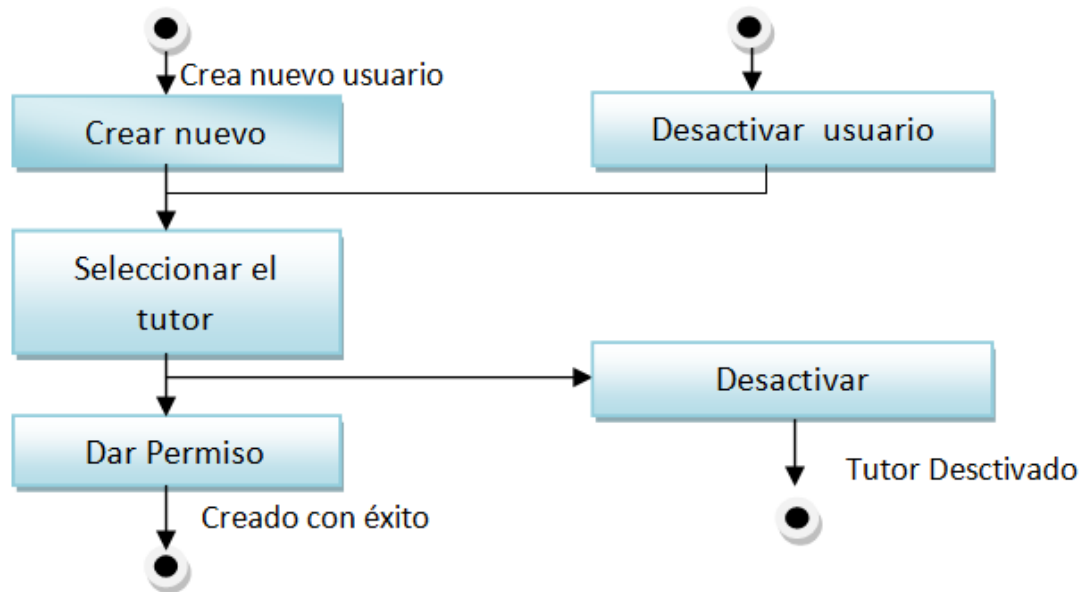


Ilustración 32: Diagrama de Estado Creación de Usuario

Fuente: Los Autores

Prorrogas

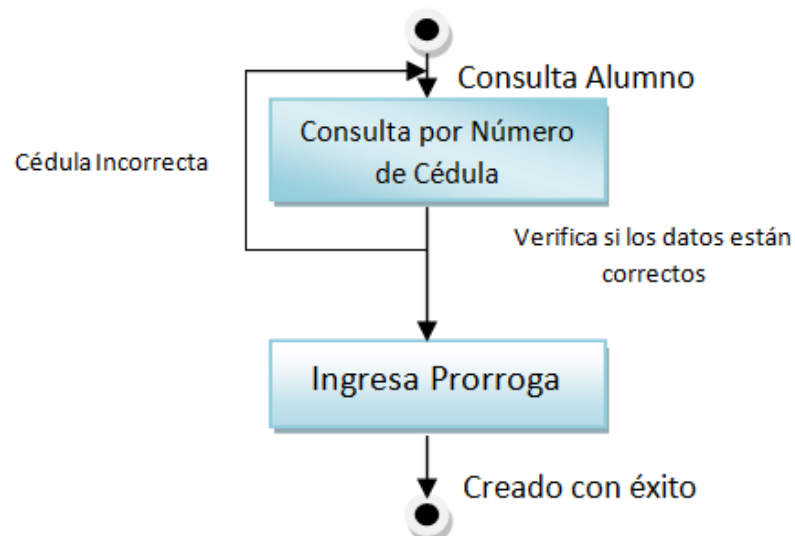


Ilustración 33: Diagrama de Estado Creación de Prorrogas

Fuente: Los Autores

Monitoreo

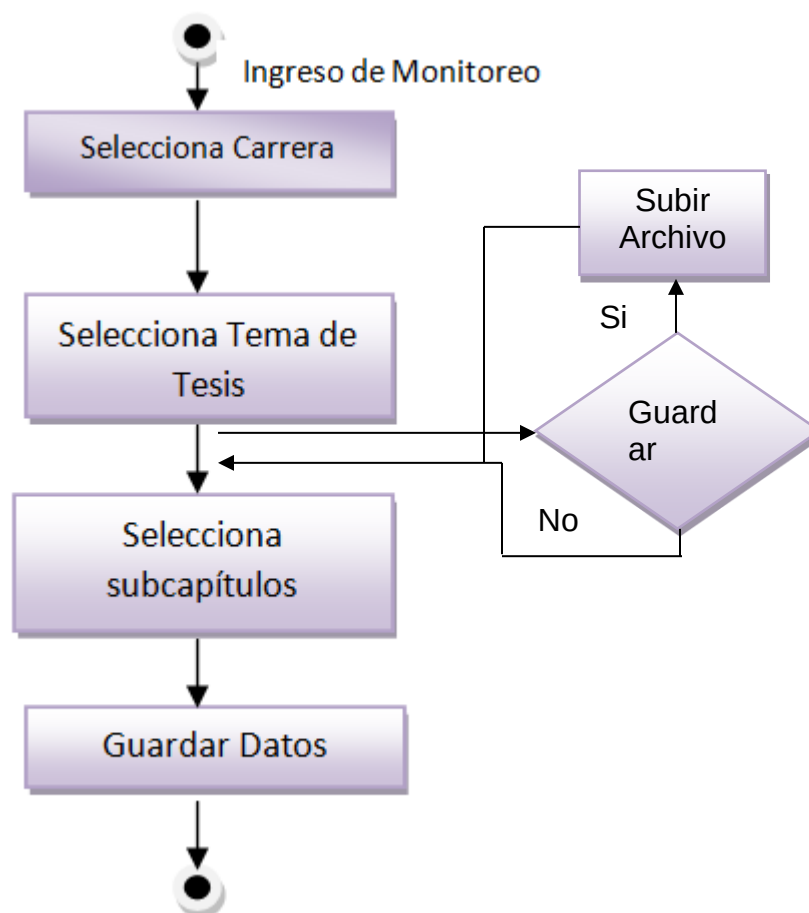


Ilustración 34: Diagrama de Estado Creación de Monitoreo

Fuente: Los Autores

3.2.6 DIAGRAMA DE SECUENCIA

El diagrama de secuencia representa la forma de cómo el usuario, se comunica a través de las clases, métodos, interactuando a través de la interfaz del usuario en la que se inician los objetos que intervienen en el funcionamiento del sistema web.

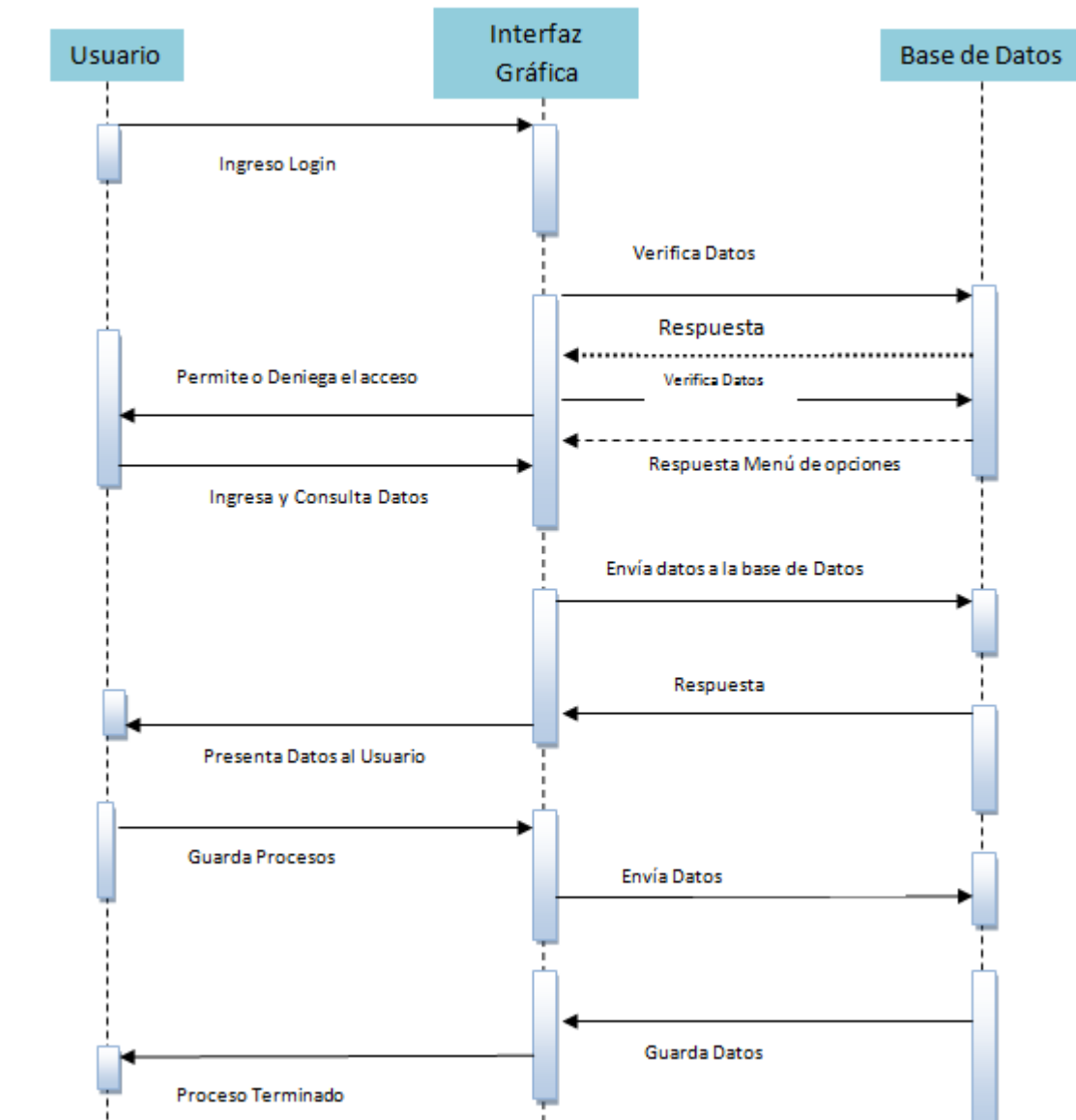


Ilustración 35: Diagrama de Secuencia

Fuente: Los Autores

3.2.7 DIAGRAMA DE CLASES

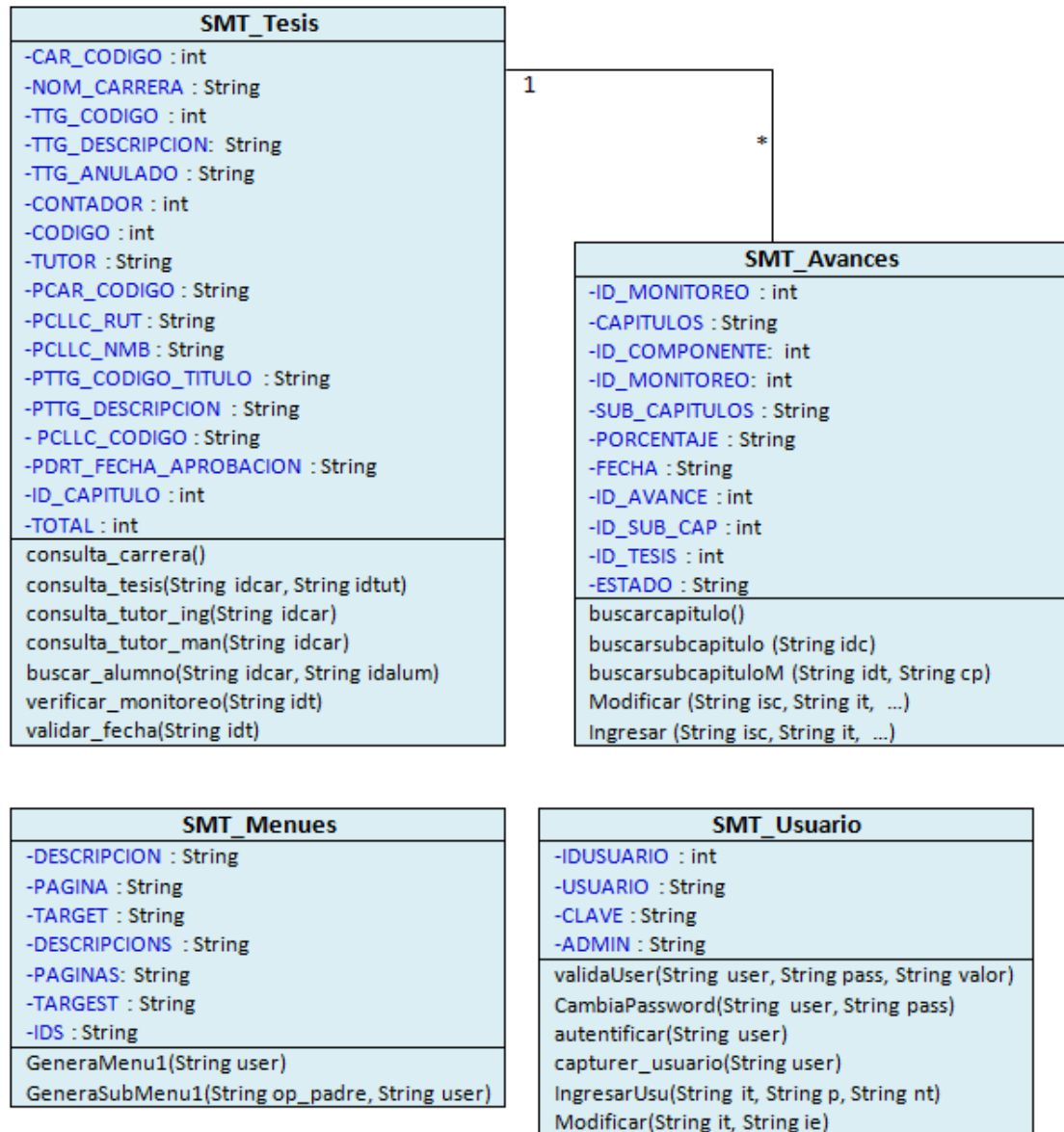


Ilustración 36: Diagrama de Secuencia

Fuente: Los Autores

Detalle Diagrama de Clase

Clase	Descripción
Tesis	Consulta de Tesis Activas
Avances	Creación de Seguimiento de Tesis
Menú	Funcionalidades para los usuarios.
Usuarios	Crear y Desactivación de los Usuarios.

Tabla 12: Detalle del Diagrama de Clases

Fuente: Los Autores

3.2.8 ANÁLISIS FUNCIONAL

Las partes principales del proyecto son:

Ingreso al sistema desarrollado en Eclipse que permite el ingreso al usuario y el uso de las diferentes opciones constan dentro del sistema dependiendo del usuario.

Para la Opción Usuario, el que estará compuesto de las siguientes opciones:

- Crear Usuarios
- Activación y Desactivación

Para la Opción Prorroga, el que estará compuesto de las siguientes opciones:

- Crear Prorroga

Para la Opción Monitoreo, el que estará compuesto de las siguientes opciones:

- Creación y Consulta del Seguimiento

Para la Opción Reporte, el que estará compuesto de las siguientes opciones:

- Generación de Reportes

Procedimiento de Usuarios

Responsable	Actividad
Secretaría	Ingresa en la Opción Usuarios Selecciona el tutor que va a desactivar o activar para poder ingresar al sistema web.

Tabla 13: Tabla de Procedimiento de Usuarios

Fuente: Los Autores

Procedimiento de Prorroga

Responsable	Actividad
Secretaría	Consulta a los alumnos por el número de cédula. Observa los datos del alumno. Crea a los alumnos la solicitud de Prorroga.

Tabla 14: Tabla de Procedimiento de Prorroga

Fuente: Los Autores

Procedimiento de Reporte

Responsable	Actividad
Secretaría	Puede Seleccionar prorroga, tesis en curso, tesis terminadas y consultar para poder ver la información.
Director de Carrera	Puede Seleccionar prorroga, tesis en curso, tesis terminadas. Información de los tutores de cuantas tesis tiene a su cargo. Consulta para poder ver la información.
Director de Trabajo de Tesis	Puede Seleccionar prorroga, tesis en curso, tesis terminadas. Información de los seguimientos de tesis . Consulta para poder ver la información.

Tabla 15: Tabla de Procedimiento de Reporte

Fuente: Los Autores

Procedimiento de Monitoreo

Responsable	Actividad
Director de Trabajo de Tesis	Selecciona la Carrera Selecciona el Tema de Tesis a Monitorear. Se presenta la información de Capítulos para ser seleccionados los que ya se han avanzado con su respectiva fecha de avance. Guarda el seguimiento de Tesis. Consulta de seguimiento de Tesis.

Tabla 16 : Tabla de Procedimiento de Monitoreo

Fuente: Los Autores

3.3 DISEÑO DE ARQUITECTURA

3.3.1 CAPA DE BASE DE DATOS

3.3.1.1 DIAGRAMA RELACIONAL

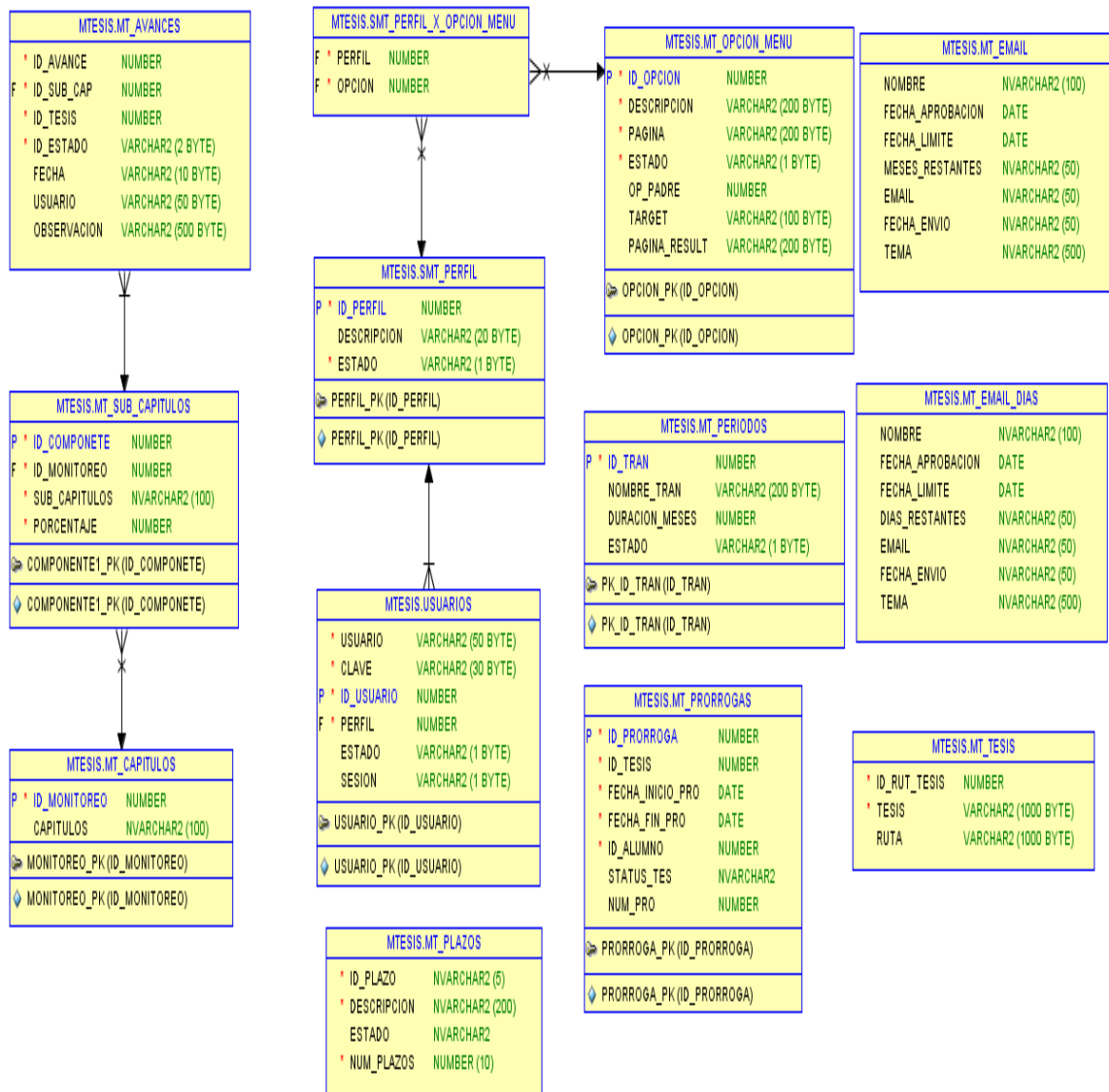


Ilustración 37: Diagrama Relacional del Sistema

Fuente: Los Autores

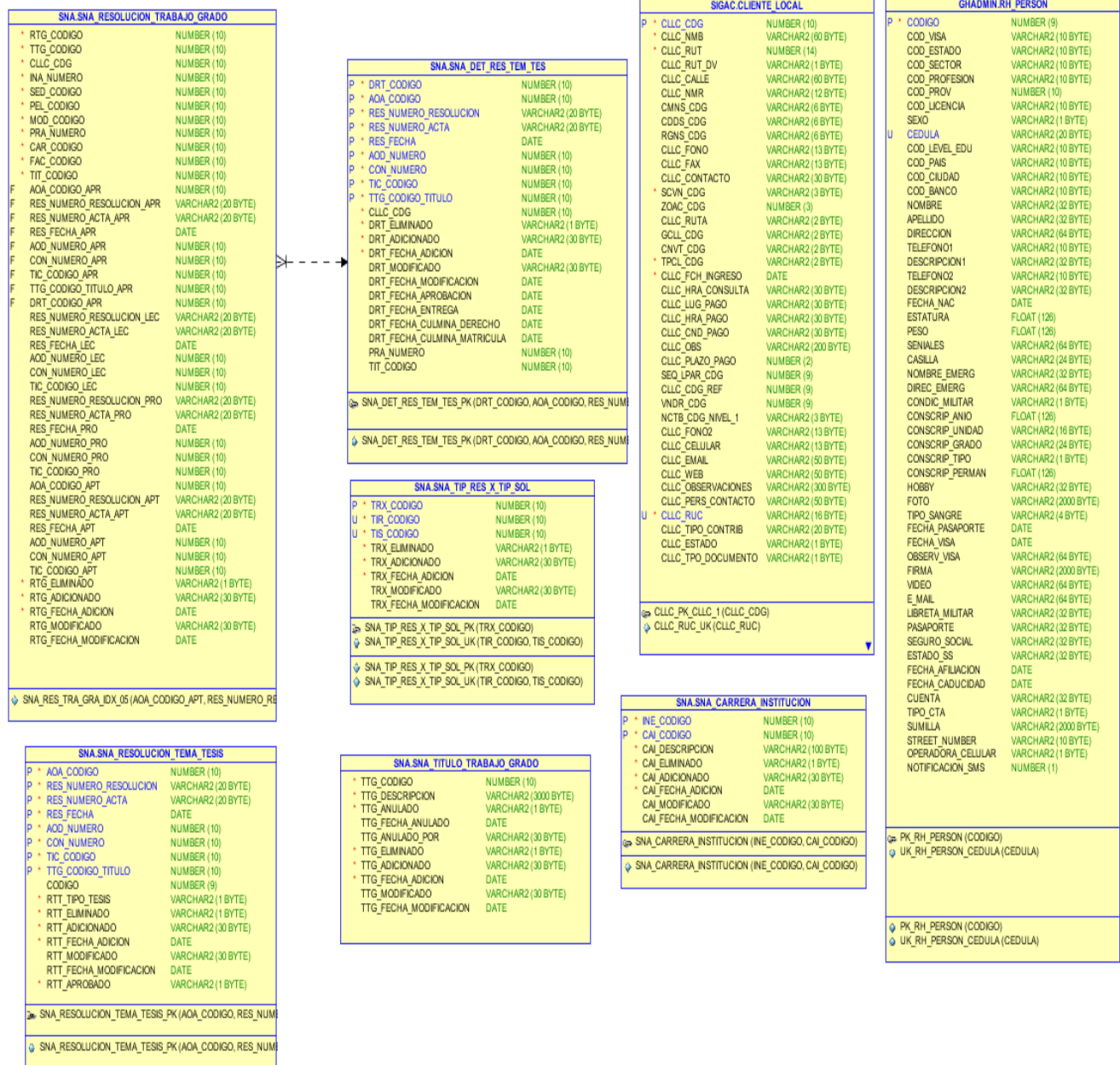


Ilustración 38: Diagrama Relacional consultada a la base de la Universidad Politécnica Salesiana

Fuente: Los Autores

3.3.1.2 DIAGRAMA ENTIDAD - RELACIÓN

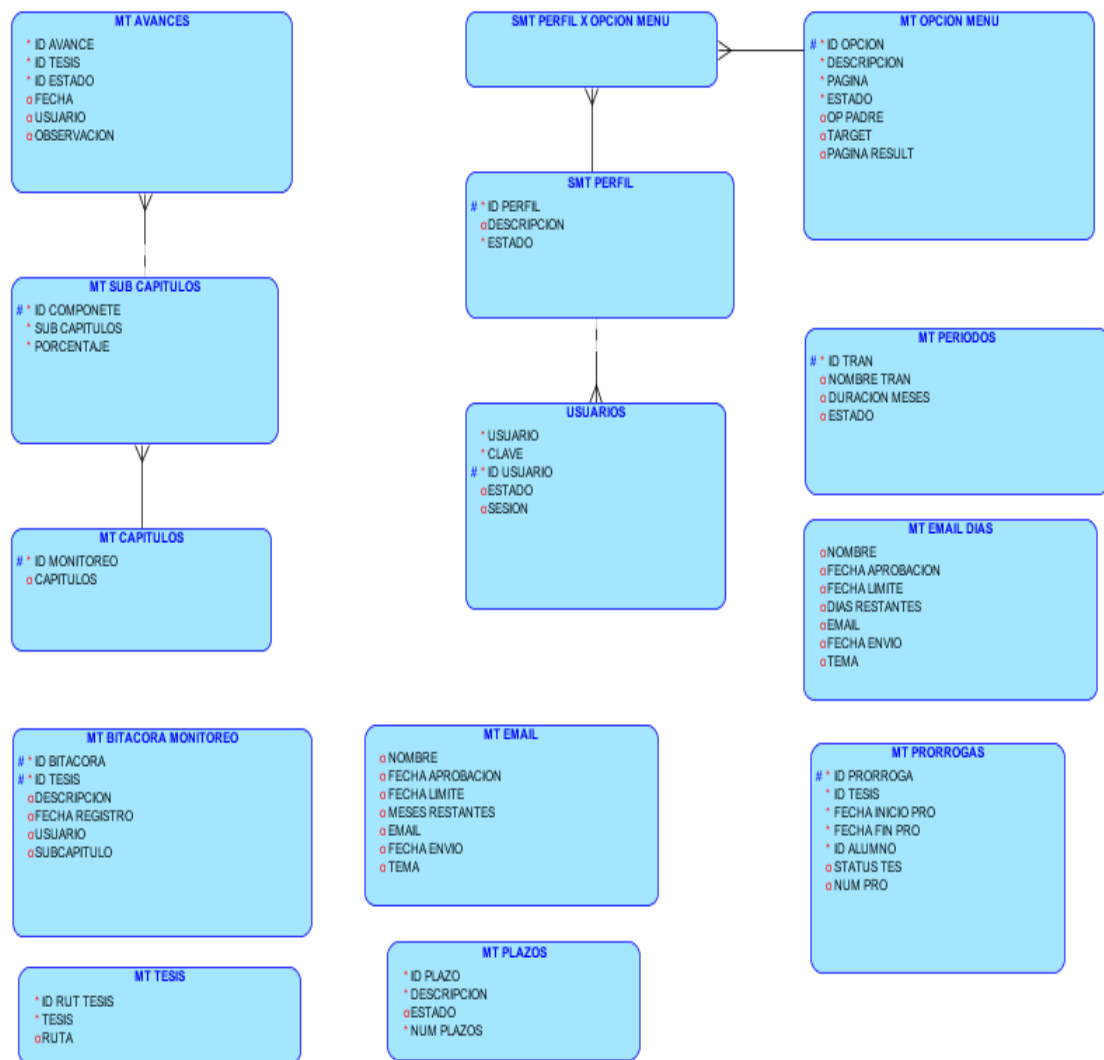


Ilustración 39: Diagrama Entidad Relación del Sistema

Fuente: Los Autores

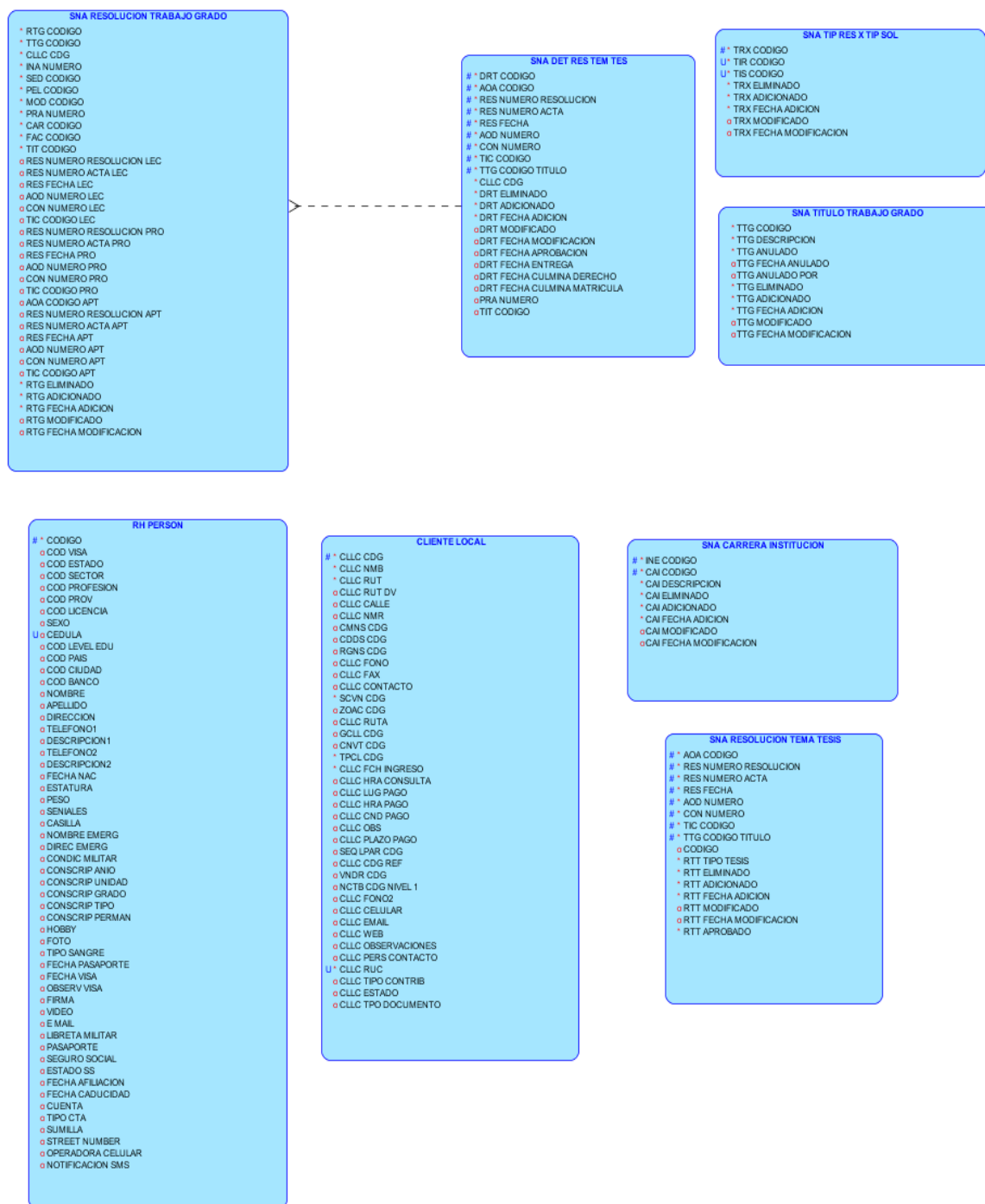


Ilustración 40: Diagrama Entidad Relación consultada a la base de la Universidad Politécnica Salesiana
Fuente: Los Autores

Creación de esquema

Para este sistema se realiza la creación del esquema del usuario en la base de datos Oracle 10 g.

```
CREATE TABLESPACE "MTESIS" LOGGING
```

```
DATAFILE 'C:\oraclexe\oradata\XE\mtesis01.dbf' SIZE 100M
```

```
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
```

--CREA EL USUARIO

```
CREATE USER MTESIS IDENTIFIED BY mtesis DEFAULT TABLESPACE MTESIS;
```

-- OTORGA LOS PERMISOS NECESARIOS

```
GRANT "CONNECT" TO "MTESIS";
```

```
GRANT "RESOURCE" TO "MTESIS";
```

```
GRANT ALTER, CREATE, DROP ANY INDEX TO "MTESIS";
```

```
GRANT ALTER, CREATE, DROP ANY SEQUENCE TO "MTESIS";
```

```
GRANT ALTER, CREATE, DROP, INSERT ,DELETE, SELECT ANY TABLE TO "MTESIS";
```

```
GRANT ALTER, CREATE, DROP ANY TRIGGER TO "MTESIS";
```

```
GRANT CREATE ANY SYNONYM TO "MTESIS";
```

```
GRANT CREATE, DROP ANY VIEW TO "MTESIS";
```

```
GRANT CREATE PROCEDURE TO "MTESIS";
```

```
GRANT CREATE PUBLIC SYNONYM TO "MTESIS";
```

```
GRANT QUERY REWRITE TO "MTESIS";
```

```
GRANT UNLIMITED TABLESPACE TO "MTESIS";
```

Detalle de la creación del esquema:

Crear un tablespace en la ruta 'C:\oraclexe\oradata\XE\', cuyo nombre definido es mtesis01

```
CREATE          TABLESPACE          "MTESIS"          LOGGING          DATAFILE  
'C:\oraclexe\oradata\XE\mtesis01L.dbf' SIZE 100M
```

Crear el usuario MTESIS con su respectiva contraseña y asignarlo al tablespace previamente creado.

```
CREATE USER MTESIS IDENTIFIED BY mtesis DEFAULT TABLESPACE;
```

*Dar privilegios de conexión al usuario mtesis.

3.3.1.2 DEFINICIÓN DE LAS TABLAS DE SISTEMA

A continuación se detallan las tablas de SMT-Web-Server (SISTEMA DE MONITOREO DE TESIS):

COLUMNA	DESCRIPCIÓN
USUARIOS	Contiene los usuarios que podrán acceder al sistema con sus respectivas claves.
MT_OPCION_MENU	Contiene las opciones que se mostrarán en el menú de la página web.
SMT_PERFIL	Consta de los perfiles de los usuarios.
MT_PRORROGAS	Consta de los datos de las prórrogas solicitadas por los estudiantes.
SMT_PERFIL_X_OPCION_MENU	Contiene las asignaciones de menú correspondientes a los perfiles.
MT_CAPITULOS	Consta de los capítulos correspondientes al monitoreo de las tesis.
MT_SUB_CAPITULOS	Consta de los subcapítulos correspondientes al monitoreo de las tesis.
MT_AVANCES	Contiene los avances por tesis.
MT_PERIODOS	Contiene los periodos válidos para tesis y prórrogas.
MT_PLAZOS	Contiene el número de plazos permitidos.

Tabla 17: Tablas de la base

Fuente: Los Autores

▪ **Tabla Usuarios**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
ID_USUARIO	NUMBER	Primary Key Not Null	Código del usuario que corresponde al de la tabla de empleados (RHPERSON)
USUARIO	VARCHAR2(50)	Not Null	Nombre del usuario que debe corresponder a la primera inicial de su nombre, su segundo apellido y la primera inicial de su segundo apellido.
CLAVE	VARCHAR2(30)	Not Null	Clave de usuario
PERFIL	NUMBER	Foreing Key Not Null	Contiene código del perfil con que va a ingresar el usuario.
ESTADO	VARCHAR2(1)		Eliminado Lógico 'T' si es usuario valido estará con 'A'
SESION	VARCHAR2(1)		Indica si la sesión esta activa.

Tabla 18: Tabla de Usuarios

Fuente: Los Autores

▪ **Tabla MT_OPCION_MENU**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
ID_OPCION	NUMBER	Primary Key Not Null	Id que identifica la opción
DESCRIPCION	VARCHAR2(200)	Not Null	Descripción que identifica el menú
PAGINA	VARCHAR2(200)	Not Null	Página a la que referenciara esta opción en el menú
ESTADO	VARCHAR2(1)		'Si está activo no
OP_PADRE	NUMBER		Opción padre de submenú
TARGET	VARCHAR2(100)		El nombre del frame en el cual se va a enviar la página resultante
PAGINA_RESULT	VARCHAR2(200)		Página resultante en caso de que actúe como intermedia

Tabla 19: Tabla de Opciones del Menú

Fuente: Los Autores

▪ **Tabla SMT_PERFIL**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
ID_PERFIL	NUMBER	Primary Key Not Null	Id de tipo de perfil
DESCRIPCION	VARCHAR2(20)	Not Null	Descripción del perfil
ESTADO	VARCHAR2(1)	Not Null	Si el perfil se encuentra activo o no

Tabla 20 : Tabla de Perfil

Fuente: Los Autores

▪ **Tabla MT_PRORROGAS**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
ID_PRORROGA	NUMBER	Primary Key Not Null	Secuencia De Prorroga
ID_TESIS	NUMBER	Not Null	Secuencia De Prorroga
FECHA_INICIO_PRO	DATE	Not Null	Fecha De Inicio De La Prorroga
FECHA_FIN_PRO	DATE	Not Null	Fehca Cuando Culmina La Prorroga
ID_ALUMNO	NUMBER	Not Null	Secuencia Del Alumno
STATUS_TES	NVARCHAR2(1)		Eliminado Logico A=Activo , I=Inactivo Para Tesis
NUM_PRO	NUMBER		Número De Prorrogas

Tabla 21: Tabla de Prorrogas

Fuente: Los Autores

▪ **Tabla SMT_PERFIL_X_OPCION_MENU**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
PERFIL	NUMBER	Foreing Key Not Null	Id de tipo de perfil
OPCION	NUMBER	Foreing Key Not Null	Id que identifica la opción

Tabla 22: Tabla de Perfil para Opciones del Menú

Fuente: Los Autores

▪ **Tabla MT_CAPITULOS**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
ID_MONITOREO	NUMBER	Primary Key Not Null	Id del capitulo
CAPITULOS	NVARCHAR2(100)	Not Null	Descripción del Capítulo

Tabla 23: Tabla de Capítulos

Fuente: Los Autores

▪ **Tabla MT_SUB_CAPITULOS**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
ID_COMPONENTE	NUMBER	Primary Key Not Null	Id del subcapítulo
ID_MONITOREO	NUMBER	Foreing Key Not Null	Id del capítulo al que pertenece
SUB_CAPITULOS	NVARCHAR2(100)	Not Null	Descripción del subcapítulo
PORCENTAJE	NUMBER	Not Null	Valor dado en porcentajes que suma cada subcapítulo

Tabla 24: Tabla de Subcapítulos

Fuente: Los Autores

▪ **Tabla MT_AVANCES**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
ID_AVANCE	NUMBER	Not Null	Id secuencial de seguimiento de tesis.
ID_SUB_CAP	NUMBER	Primary Key Foreing Key Not Null	Id del subcapítulo al que pertenece este seguimiento
ID_TESIS	NUMBER	Primary Key Not Null	Id de la tesis a la que pertenece el seguimiento de este subcapítulo
ID_ESTADO	VARCHAR2(2)		Indica 'SI' al está realizado dicho subcapítulo y 'NO' al no estar desarrollado
FECHA	VARCHAR2(10)	Not Null	La fecha en la que se realizó el seguimiento
USUARIO	VARCHAR2(50)	Not Null	

Tabla 25: Tabla de Avances

Fuente: Los Autores

▪ **Tabla MT_TESIS**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
ID_RUT_TESIS	NUMBER	Primary Key Not Null	Id secuencial de rutas.
RUTA	VARCHAR2(1000)	Not Null	Ruta lógica del archivo
TESIS	VARCHAR2(1000)	Not Null	Id de la tesis a la que pertenece el archivo.

Tabla 26: Tabla de Tesis

Fuente: Los Autores

▪ **Tabla MT_PERIODOS**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
ID_TRAN_NUMBER	NUMBER	Primary Key Not Null	Id secuencial de trámites.
NOMBRE_TRAN	VARCHAR2(200)	Not Null	Nombre del trámite
DURACION_MESES	NUMBER	Not Null	Tiempo
ESTADO	VARCHAR2(1)	Not Null	Activo o inactivo

Tabla 27: Tabla de Periodos

Fuente: Los Autores

▪ **Tabla MT_PLAZOS**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
ID_PLAZO	VARCHAR2(5)	Primary Key	Id secuencial.
DESCRIPCION	VARCHAR2(200)	Not Null	DESCRIPCIÓN
ESTADO	VARCHAR2(1)	Not Null	Activo o inactivo
NUM_PLAZOS	NUMBER(10)	Not Null	Número de plazos

Tabla 28: Tabla de Plazos

Fuente: Los Autores

A continuación se detallan las tablas consultadas a la base de la Universidad Politécnica Salesiana:

▪ **Tabla SNA_RESOLUCION_TRABAJO_GRADO**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
RTG_CODIGO	NUMBER(10)	Primary Key	Código secuencial de la tabla
TTG_CODIGO	NUMBER(10)		
CLLC_CDG	NUMBER(10)		Código secuencial de los temas
INA_NUMERO	NUMBER(10)		
SED_CODIGO	NUMBER(10)		Código del alumno
PEL_CODIGO	NUMBER(10)		
MOD_CODIGO	NUMBER(10)		Código de la inscripción del alumno
PRA_NUMERO	NUMBER(10)		
CAR_CODIGO	NUMBER(10)		Código de la sede de la inscripción del alumno
FAC_CODIGO	NUMBER(10)		
TIT_CODIGO	NUMBER(10)		Código del periodo lectivo
AOA_CODIGO_APR	NUMBER(10)		
RES_NUMERO_RESOLUCION_APR	VARCHAR2(20)		Código de la modalidad
RES_NUMERO_ACTA_APR	VARCHAR2(20)		
RES_FECHA_APR	DATE		Código del proyecto, malla, mención
AOD_NUMERO_APR	NUMBER(10)		
CON_NUMERO_APR	NUMBER(10)		Código de la carrera
TIC_CODIGO_APR	NUMBER(10)		
TTG_CODIGO_TITULO_APR	NUMBER(10)		Código de la facultad
DRT_CODIGO_APR	NUMBER(10)		
RES_NUMERO_RESOLUCION_LEC	VARCHAR2(20)		Código del título
RES_NUMERO_ACTA_LEC	VARCHAR2(20)		
RES_FECHA_LEC	DATE		Número de antecedente

AOD_NUMERO_LEC	NUMBER(10)		Número de la resolución
CON_NUMERO_LEC	NUMBER(10)		
TIC_CODIGO_LEC	NUMBER(10)		Número de acta
RES_NUMERO_RESOLUCION_PRO	VARCHAR2(20)		
RES_NUMERO_ACTA_PRO	VARCHAR2(20)		Fecha de la resolución
RES_FECHA_PRO	DATE		
AOD_NUMERO_PRO	NUMBER(10)		Número de acta del orden del día
CON_NUMERO_PRO	NUMBER(10)		
TIC_CODIGO_PRO	NUMBER(10)		Número de convocatoria
AOA_CODIGO_APT	NUMBER(10)		
RES_NUMERO_RESOLUCION_APT	VARCHAR2(20)		Indica el código del consejo
RES_NUMERO_ACTA_APT	VARCHAR2(20)		
RES_FECHA_APT	DATE		Código del título de aprobación del tema de tesis
AOD_NUMERO_APT	NUMBER(10)		
CON_NUMERO_APT	NUMBER(10)		Código secuencial de la tabla sna_det_res_tem_tes
TIC_CODIGO_APT	NUMBER(10)		
RTG_ELIMINADO	VARCHAR2(1)	Not Null	Número de la resolución lectura del tema
RTG_ADICIONADO	VARCHAR2(30)	Not Null	
RTG_FECHA_ADICION	DATE	Not Null	Número de la acta de la resolución lectura del tema
RTG_MODIFICADO	VARCHAR2(30)		
RTG_FECHA_MODIFICACION	DATE		Fecha de la resolución lectura del tema

Tabla 29: Tabla de SNA_RESOLUCION_TRABAJO_GRADO

Fuente: Los Autores

▪ **Tabla SNA_DET_RES_TEM_TES**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
DRT_CODIGO	NUMBER(10)	Primary Key	Numero secuencial del detalle de tema de tesis
AOA_CODIGO	NUMBER(10)	Not null	
RES_NUMERO_RESOLUCION	VARCHAR2(20)	Not null	Número del antecedente
RES_NUMERO_ACTA	VARCHAR2(20)	Not null	
RES_FECHA	DATE	Not null	Número de la resolución.
AOD_NUMERO	NUMBER(10)	Not null	
CON_NUMERO	NUMBER(10)	Not null	Número de la acta de la resolución.
TIC_CODIGO	NUMBER(10)	Not null	
TTG_CODIGO_TITULO	NUMBER(10)	Not null	Fecha de la resolución.
CLLC_CDG	NUMBER(10)	Not null	
DRT_ELIMINADO	VARCHAR2(1)	Not null	Número del acta del orden del día.
DRT_ADICIONADO	VARCHAR2(30)	Not null	
DRT_FECHA_ADICION	DATE	Not null	Número de convocatoria.
DRT_MODIFICADO	VARCHAR2(30)		
DRT_FECHA_MODIFICACION	DATE		Código del tipo de consejo.
DRT_FECHA_APROBACION	DATE		
DRT_FECHA_ENTREGA	DATE		Código del tema de tesis
DRT_FECHA_CULMINA_DERECHO	DATE		
DRT_FECHA_CULMINA_MATRICULA	DATE		Código del alumno

PRA_NUMERO	NUMBER(10)		
TIT_CODIGO	NUMBER(10)		Eliminado lógico de la convocatoria. N=no, S=si.

Tabla 30: Tabla SNA_DET_RES_TEM_TES

Fuente: Los Autores

▪ **Tabla SNA_CARRERA_INSTITUCION**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
INE_CODIGO	NUMBER(10)	Primary key	Código de la Institución Educativa.
CAI_CODIGO	NUMBER(10)	Not null	Código de la Carrera de la Institución Educativa.
CAI_DESCRIPCION	VARCHAR2(100)	Not null	Descripción de la Carrera de la Institución.
CAI_ELIMINADO	VARCHAR2(1)	Not null	Eliminado Lógico.
CAI_ADICIONADO	VARCHAR2(30)	Not null	Usuario que adiciona el registro.
CAI_FECHA_ADICION	DATE	Not null	Fecha en la que se adiciona el registro.
CAI_MODIFICADO	VARCHAR2(30)		Usuario que modifica el registro.
CAI_FECHA_MODIFICACION	DATE		Fecha en la que se modifica el registro.

Tabla 31: Tabla de Sna_Carrera_Institucion

Fuente: Los Autores

▪ **Tabla CLIENTE_LOCAL**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
CLLC_CDG	NUMBER(10)	Primary Key	
CLLC_NMB	VARCHAR2(60)	Not null	
CLLC_RUT	NUMBER(14)	Not null	
CLLC_RUT_DV	VARCHAR2(1)		
CLLC_CALLE	VARCHAR2(60)		
CLLC_NMR	VARCHAR2(12)		
CMNS_CDG	VARCHAR2(6)		
CDDS_CDG	VARCHAR2(6)		
RGNS_CDG	VARCHAR2(6)		
CLLC_FONO	VARCHAR2(13)		
CLLC_FAX	VARCHAR2(13)		
CLLC_CONTACTO	VARCHAR2(30)		
SCVN_CDG	VARCHAR2(3)	Not null	
ZOAC_CDG	NUMBER(3)		
CLLC_RUTA	VARCHAR2(2)		
GCLL_CDG	VARCHAR2(2)		
CNVT_CDG	VARCHAR2(2)		
TPCL_CDG	VARCHAR2(2)	Not null	
CLLC_FCH_INGRESO	DATE	Not null	
CLLC_HRA_CONSULTA	VARCHAR2(30)		
CLLC_LUG_PAGO	VARCHAR2(30)		

CLLC_HRA_PAGO	VARCHAR2(30)		
CLLC_CND_PAGO	VARCHAR2(30)		
CLLC_OBS	VARCHAR2(200)		
CLLC_PLAZO_PAGO	NUMBER(2)		
SEQ_LPAR_CDG	NUMBER(9)		
CLLC_CDG_REF	NUMBER(9)		
VNDR_CDG	NUMBER(9)		
NCTB_CDG_NIVEL_1	VARCHAR2(3)		
CLLC_FONO2	VARCHAR2(13)		
CLLC_CELULAR	VARCHAR2(13)		
CLLC_EMAIL	VARCHAR2(50)		
CLLC_WEB	VARCHAR2(50)		
CLLC_OBSERVACIONES	VARCHAR2(300)		
CLLC_PERS_CONTACTO	VARCHAR2(50)		
CLLC_RUC	VARCHAR2(16)	Not null	
CLLC_TIPO_CONTRIB	VARCHAR2(20)		
CLLC_ESTADO	VARCHAR2(1)		
CLLC_TPO_DOCUMENTO	VARCHAR2(1)		

Tabla 32: Tabla de Cliente Local

Fuente: Los Autores

▪ **Tabla RH_PERSON**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
CODIGO	NUMBER(9)	Primary Key	
COD_VISA	VARCHAR2(10)		
COD_ESTADO	VARCHAR2(10)		
COD_SECTOR	VARCHAR2(10)		
COD_PROFESION	VARCHAR2(10)		
COD_PROV	NUMBER(10)		
COD_LICENCIA	VARCHAR2(10)		
SEXO	VARCHAR2(1)		
CEDULA	VARCHAR2(20)		
COD_LEVEL_EDU	VARCHAR2(10)		
COD_PAIS	VARCHAR2(10)		
COD_CIUADAD	VARCHAR2(10)		
COD_BANCO	VARCHAR2(10)		
NOMBRE	VARCHAR2(32)		
APELLIDO	VARCHAR2(32)		
DIRECCION	VARCHAR2(64)		
TELEFONO1	VARCHAR2(10)		
DESCRIPCION1	VARCHAR2(32)		
TELEFONO2	VARCHAR2(10)		
DESCRIPCION2	VARCHAR2(32)		
FECHA_NAC	DATE		

ESTATURA	FLOAT		
PESO	FLOAT		
SENIALES	VARCHAR2(64)		
CASILLA	VARCHAR2(24)		
NOMBRE_EMERG	VARCHAR2(32)		
DIREC_EMERG	VARCHAR2(64)		
CONDIC_MILITAR	VARCHAR2(1)		
CONSCRIP_ANIO	FLOAT		
CONSCRIP_UNIDAD	VARCHAR2(16)		
CONSCRIP_GRADO	VARCHAR2(24)		
CONSCRIP_TIPO	VARCHAR2(1)		
CONSCRIP_PERMAN	FLOAT		
HOBBY	VARCHAR2(32)		
FOTO	VARCHAR2(2000)		
TIPO_SANGRE	VARCHAR2(4)		
FECHA_PASAPORTE	DATE		
FECHA_VISA	DATE		
OBSERV_VISA	VARCHAR2(64)		
FIRMA	VARCHAR2(2000)		
VIDEO	VARCHAR2(64)		
E_MAIL	VARCHAR2(64)		
LIBRETA_MILITAR	VARCHAR2(32)		
PASAPORTE	VARCHAR2(32)		

SEGURO_SOCIAL	VARCHAR2(32)		
ESTADO_SS	VARCHAR2(32)		
FECHA_AFILIACION	DATE		
FECHA_CADUCIDAD	DATE		
CUENTA	VARCHAR2(32)		
TIPO_CTA	VARCHAR2(1)		
SUMILLA	VARCHAR2(2000)		
STREET_NUMBER	VARCHAR2(10)		
OPERADORA_CELULAR	VARCHAR2(1)		
NOTIFICACION_SMS	NUMBER(1)		

Tabla 33: Tabla de RH_PERSON

Fuente: Los Autores

▪ **Tabla SNA_RESOLUCION_TEMA_TESIS**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
AOA_CODIGO	NUMBER(10)	Primary Key	Número del Antecedente
RES_NUMERO_RESOLUCION	VARCHAR2(20)	Not Null	Número de la resolución.
RES_NUMERO_ACTA	VARCHAR2(20)	Not Null	Número del acta de la resolución.
RES_FECHA	DATE	Not Null	Fecha de la resolución.
AOD_NUMERO	NUMBER(10)	Not Null	Número del acta del orden del día.

CON_NUMERO	NUMBER(10)	Not Null	Número de convocatoria.
TIC_CODIGO	NUMBER(10)	Not Null	Código del tipo de consejo.
TTG_CODIGO_TITULO	NUMBER(10)	Not Null	Código del tema de tesis
CODIGO	NUMBER(9)		Código del director de tesis asignado
RTT_TIPO_TESIS	VARCHAR2(1)	Not Null	Tipo de tesis
RTT_ELIMINADO	VARCHAR2(1)	Not Null	Eliminado lógico de la convocatoria. N=no, S=si.
RTT_ADICIONADO	VARCHAR2(30)	Not Null	Identifica a la persona que realizó una adición en la tabla.
RTT_FECHA_ADICION	DATE	Not Null	Describe la fecha en la que se realizó la adición del registro en la tabla.
RTT_MODIFICADO	VARCHAR2(30)		Identifica a la persona que realizó una modificación al registro.
RTT_FECHA_MODIFICACION	DATE		Describe la fecha en la que se realizó la modificación del registro en la tabla.
RTT_APROBADO	VARCHAR2(1)	Not Null	Aprobado el Tema de Tesis A = Aprobado N = Negado

Tabla 34: Tabla SNA_RESOLUCION_TEMA_TESIS

Fuente: Los Autores

▪ **Tabla SNA_TIP_RES_X_TIP_SOL**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
TRX_CODIGO	NUMBER(10)	Primary Key	Número secuencial del tipo de resolución por solicitud
TIR_CODIGO	NUMBER(10)	Not null	Número del tipo de resolución
TIS_CODIGO	NUMBER(10)	Not null	Número del tipo de solicitud
TRX_ELIMINADO	VARCHAR2(1)	Not null	Eliminado lógico de la convocatoria. N=no, S=si.
TRX_ADICIONADO	VARCHAR2(30)	Not null	Identifica a la persona que realizó una adición en la tabla.
TRX_FECHA_ADICION	DATE	Not null	Describe la fecha en la que se realizó la adición del registro en la tabla.
TRX_MODIFICADO	VARCHAR2(30)		Identifica a la persona que realizó una modificación al registro.
TRX_FECHA_MODIFICACION	DATE		Describe la fecha en la que se realizó la modificación del registro en la tabla.

Tabla 35: Tabla de RH_PERSON

Fuente: Los Autores

▪ **Tabla SNA_RESOLUCION_TEMA_TESIS**

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
AOD_NUMERO	NUMBER(10)	Primary Key	Número de la acta del orden del día.
CON_NUMERO	NUMBER(10)	Not Null	Número de convocatoria.
TIC_CODIGO	NUMBER(10)	Not Null	Código del tipo de consejo.

TTG_CODIGO_TITULO	NUMBER(10)	Not Null	Código del tema de tesis
CODIGO	NUMBER(9)		Código del director de tesis asignado
RTT_TIPO_TESIS	VARCHAR2(1)	Not Null	Tipo de tesis
RTT_ELIMINADO	VARCHAR2(1)	Not Null	Eliminado lógico de la convocatoria. N=no, S=si.
RTT_ADICIONADO	VARCHAR2(30)	Not Null	Identifica a la persona que realizó una adición en la tabla.
RTT_FECHA_ADICION	DATE	Not Null	Describe la fecha en la que se realizó la adición del registro en la tabla.
RTT_MODIFICADO	VARCHAR2(30)		Identifica a la persona que realizó una modificación al registro.
RTT_FECHA_MODIFICACION	DATE		Describe la fecha en la que se realizó la modificación del registro en la tabla.
RTT_APROBADO	VARCHAR2(1)	Not Null	Aprobado el Tema de Tesis A = Aprobado N = Negado

Tabla 36: Tabla de SNA_RESOLUCION_TEMA_TESIS

Fuente: Los Autores

3.3.1.3 Definición de Paquetes

El SMT Web-Server (SISTEMA DE MONITOREO DE TESIS) en su estructura contiene paquetes por cada área que abarca el sistema y en su interior se implementan procedimientos almacenados.

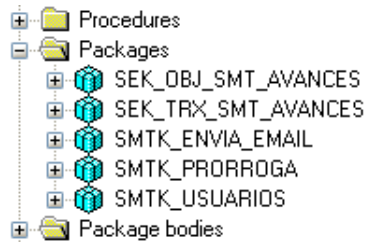


Ilustración 41: Estructura de Paquetes

Fuente: Los Autores

SEK_OBJ_SMT_AVANCES

Paquete que permite ingresar, actualizar el avance de las respectivas tesis que se están llevando a cabo por los estudiantes de la Universidad Politécnica Salesiana; y contiene los procedimientos:

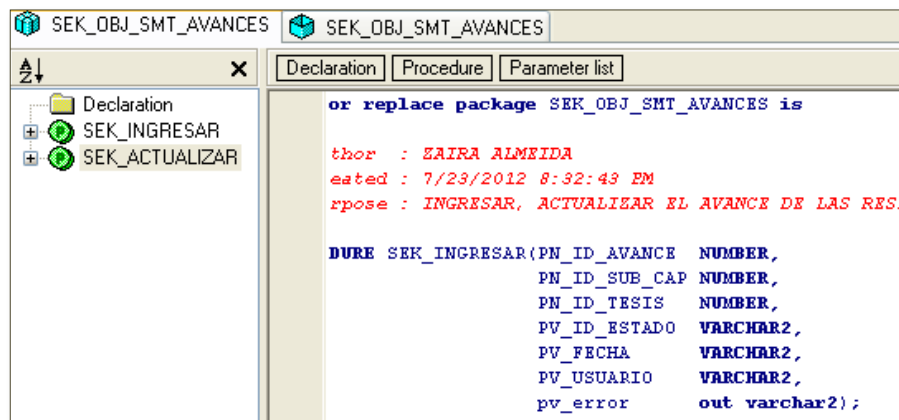


Ilustración 42: Paquete SEK_OBJ_SMT_AVANCES

Fuente: Los Autores

SEK_INGRESAR

Procedimiento que permite el ingreso de una nuevo avance en el monitoreo, solo se utiliza la primera vez que se ingresa monitoreo de una tesis, este procedimiento recibe los parámetros del procedimiento SEK_TRX_SMT_AVANCES. SEK_INGRESAR.



Código del procedimiento:

```
CREATE OR REPLACE PACKAGE BODY SEK_OBJ_SMT_AVANCES is
/******INGRESAR*****/

PROCEDURE SEK_INGRESAR(PN_ID_AVANCE NUMBER,
                        PN_ID_SUB_CAP NUMBER,
                        PN_ID_TESIS NUMBER,
                        PV_ID_ESTADO VARCHAR2,
                        PV_FECHA VARCHAR2,
                        PV_USUARIO VARCHAR2,
                        PV_OBS VARCHAR2,
                        pv_error out varchar2) IS

BEGIN
  INSERT INTO MTESIS.MT_AVANCES
  VALUES
    (PN_ID_AVANCE,
     PN_ID_SUB_CAP,
     PN_ID_TESIS,
     PV_ID_ESTADO,
     PV_FECHA,
     PV_USUARIO,
     PV_OBS);
  COMMIT;

EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    pv_error := 'Error en SEK_INGRESAR ' + sqlerrm;

END SEK_INGRESAR;
```

SEK_ACTUALIZAR

Procedimiento que permite la actualización de los avances de un monitoreo donde cada vez que el parametro de entrada pv_id_estado venga como SI se actualizará la fecha de registro del ok de dicho monitoreo.



Código del procedimiento:

```
/**ACTUALIZAR***/
PROCEDURE SEK_ACTUALIZAR(PN_ID_SUB_CAP NUMBER,
                        PN_ID_TESIS  NUMBER,
                        PV_ID_ESTADO  VARCHAR2,
                        PV_USUARIO  VARCHAR2,
                        pv_error out varchar2) IS

    LV_FECHA VARCHAR2(100);

BEGIN
    --cursor que valida si ya se ingresó monitoreo por medio de la fecha
    SELECT A.FECHA
    INTO LV_FECHA
    FROM MTESIS.MT_AVANCES A
    WHERE A.ID_TESIS = PN_ID_TESIS
    AND A.ID_SUB_CAP = PN_ID_SUB_CAP;
    --si no ha sido ingresado y viene como estado de hecho le va a asignar la fecha de día que está ingresando
    IF (LV_FECHA IS NULL) THEN
        IF (PV_ID_ESTADO = 'SI') THEN
            LV_FECHA := SYSDATE;
        END IF;
    ELSE
        LV_FECHA := LV_FECHA;
    END IF;
    --Realiza la actualización
    UPDATE MTESIS.MT_AVANCES A
    SET A.ID_ESTADO = NVL(PV_ID_ESTADO, A.ID_ESTADO),
        A.FECHA    = NVL(LV_FECHA, A.FECHA),
        A.USUARIO  = NVL(PV_USUARIO, A.USUARIO)
    WHERE A.ID_SUB_CAP = PN_ID_SUB_CAP
    AND A.ID_TESIS = PN_ID_TESIS;

    COMMIT;

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        pv_error := 'Error en SEK_ACTUALIZAR';

END SEK_ACTUALIZAR;

end SEK_OBJ_SMT_AVANCES;
```

SEK_TRX_SMT_AVANCES

Paquete que permite la validación de los parámetros que van a servir para ingresar el avance de las respectivas tesis que se están llevando a cabo por los estudiantes contiene un único procedimiento:

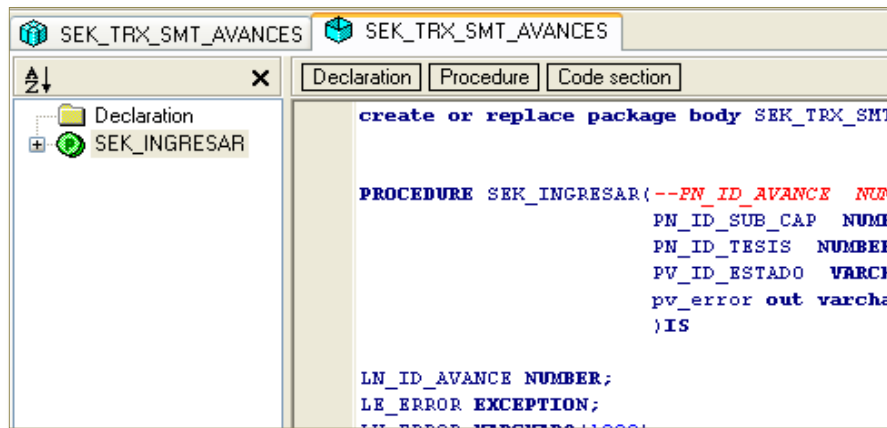


Ilustración 43: Paquete SEK_TRX_SMT_AVANCES
Fuente: Los Autores

SEK_INGRESAR

Procedimiento que permite el ingreso de un nuevo avance en el monitoreo, solo se utiliza la primera vez que se ingresa monitoreo de una tesis, utiliza la secuencia MTESIS_ID_AVANCES_SEQ para la creación de ID de avance.



Código del procedimiento:

```
create or replace package body SEK_TRX_SMT_AVANCES is

  PROCEDURE SEK_INGRESAR(PN_ID_SUB_CAP NUMBER,
                        PN_ID_TESIS  NUMBER,
                        PV_ID_ESTADO VARCHAR2,
                        PV_USUARIO VARCHAR2,
                        pv_error  out varchar2) IS

    LN_ID_AVANCE NUMBER;
    LE_ERROR EXCEPTION;
    LV_ERROR VARCHAR2(1000);
    LV_FECHA VARCHAR2(100);

  BEGIN

    --SECUENCIA
    SELECT MTESIS.MTESIS_ID_AVANCES_SEQ.NEXTVAL
    INTO LN_ID_AVANCE
    FROM DUAL;

    IF (PV_ID_ESTADO = 'NO') THEN
      LV_FECHA := '';
    ELSE
      LV_FECHA := SYSDATE;
    END IF;

    --LLAMO PAQUETE OBJ
    SEK_OBJ_SMT_AVANCES.SEK_INGRESAR(LN_ID_AVANCE,
                                      PN_ID_SUB_CAP,
                                      PN_ID_TESIS,
                                      PV_ID_ESTADO,
                                      LV_FECHA,
                                      PV_USUARIO,
                                      lv_error);

    IF LV_ERROR IS NULL THEN
      COMMIT;
    ELSE
      RAISE LE_ERROR;
    END IF;

  EXCEPTION
    WHEN LE_ERROR THEN
      ROLLBACK;
      pv_error := LV_ERROR || 'OBJ';
    WHEN OTHERS THEN
      ROLLBACK;
      pv_error := 'Error en SEK_INGRESAR TRX';

  END SEK_INGRESAR;

end SEK_TRX_SMT_AVANCES;
```

MTESIS_ID_AVANCES_SEQ

Secuencia que incrementa de 1 a 1 y se utiliza para ingresar el ID de avance.

SMTK_USUARIOS

Paquete que permite ingresar, actualizar estado de los usuarios de tipo 2, es decir tutores, además de la actualización de la clave de todos los usuarios, consta de los siguientes procedimientos almacenados:

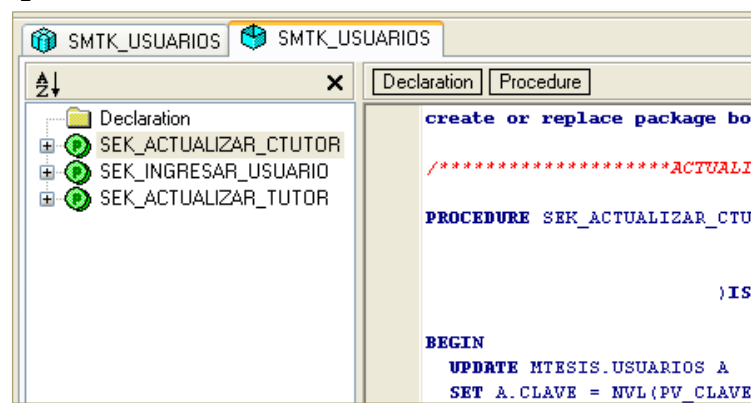


Ilustración 44: Paquete SEK_OBJ_SMT_AVANCES
Fuente: Los Autores

SEK_ACTUALIZAR_CTUTOR

Procedimiento que permite al usuario de cualquier tipo cambiar la clave para mayor seguridad.



Código del procedimiento:

```
CREATE OR REPLACE PACKAGE BODY SMTK_USUARIOS is

/*****ACTUALIZAR CLAVE DE USUARIOS*****/

PROCEDURE SEK_ACTUALIZAR_CTUTOR( PN_USUARIO NUMBER,
                                PV_CLAVE VARCHAR2,
                                pv_error out varchar2
                                )IS

BEGIN
    UPDATE MTESIS.USUARIOS A
    SET A.CLAVE = NVL(PV_CLAVE,A.CLAVE)
    WHERE A.USUARIO = PN_USUARIO AND
          A.ESTADO = 'A';

    COMMIT;

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        pv_error:= 'Error en SEK_ACTUALIZAR_CTUTOR';

END SEK_ACTUALIZAR_CTUTOR;
```

SEK_INGRESAR_USUARIO

Procedimiento que permite el ingreso de nuevos usuarios solo de tipo 2, es decir tutores.



Código del procedimiento:

```

/*****INGRESAR USUARIO TUTOR*****/

PROCEDURE SEK_INGRESAR_USUARIO( PN_USUARIO varchar2,
                                PV_CLAVE  VARCHAR2,
                                PN_IDUSUARIO NUMBER,
                                pv_error out varchar2
                                )IS

BEGIN

    INSERT INTO MTESIS.USUARIOS(USUARIO,CLAVE,ID_USUARIO,PERFIL,ESTADO,SESION) VALUES
        (PN_USUARIO,
         PV_CLAVE,
         PN_IDUSUARIO,
         2,
         'A',
         'T');

    COMMIT;

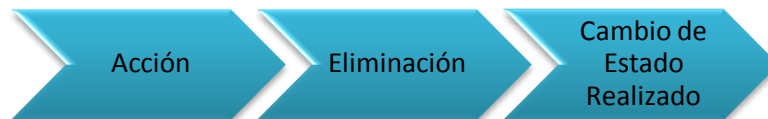
    EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        pv_error:= 'Error en SEK_INGRESAR_USUARIO';

END SEK_INGRESAR_USUARIO;

```

SEK_ACTUALIZAR_TUTOR

Procedimiento que permite realizar el eliminado lógico de los usuarios de tipo 2(tutores).



Código del procedimiento:

```

/*****ELIMINADO LOGICO DE USUARIO TUTOR*****/

PROCEDURE SEK_ACTUALIZAR_TUTOR( PN_USUARIO NUMBER,
                                PV_ID_ESTADO VARCHAR2,
                                pv_error out varchar2
                                )IS

BEGIN
    UPDATE MTESIS.USUARIOS A
    SET A.ESTADO = NVL(PV_ID_ESTADO,A.ESTADO)
    WHERE A.ID_USUARIO = PN_USUARIO AND
          A.PERFIL = 2;

    COMMIT;

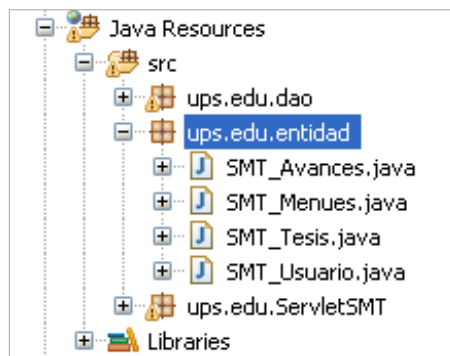
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        pv_error:= 'Error en SEK_ACTUALIZAR_TUTOR';

END SEK_ACTUALIZAR_TUTOR;
end SMTK_USUARIOS;

```

3.3.2 DAL (Data Access Layer) Capa de Acceso a Datos

En la estructura se definen las entidades de las diferentes tablas del sistema, llevamos el formato de nombre del paquete Empresa. Ocupación. Contenido, veamos:



Son cuatro entidades y las hemos dividido por la que contiene datos de avances, para la formación de menús por perfiles, datos descriptivos del tema y la entidad de datos de usuario.

Entidad SMT_Tesis

```
package ups.edu.entidad;
```

```
public class SMT_Tesis {
```

```
    //CARRERA
```

```
    private Integer CAR_CODIGO;
```

```
    private String NOM_CARRERA;
```

```
    //TESIS
```

```
    private Integer TTG_CODIGO;
```

```
    private String TTG_DESCRIPCION;
```

```
    private String TTG_ANULADO;
```

```
    private Integer Contador;
```

```
    //TUTOR
```

```
    private Integer CODIGO;
```

```
    private String TUTOR;
```

```
    //TESIS-ALUMNO
```

```
    private String pcar_codigo;
```

```
    private String pcllc_rut;
```

```
    private String pcllc_nmb;
```

```
    private String pttg_codigo_titulo;
```

```
    private String pttg_descripcion ;
```

```
    private String pcllc_codigo ;
```

```
    private String PDRT_FECHA_APROBACION;
```

```
    private Integer id_capitulo;
```

```
    private Integer total;
```

```
    public String getPcllc_codigo() {
```

```
        return pcllc_codigo;
```

```
    }
```

```
    public void setPcllc_codigo(String pcllc_codigo) {
```

```
        this.pcllc_codigo = pcllc_codigo;
```

```
    }
```

```
    public Integer getCAR_CODIGO() {
```

```
        return CAR_CODIGO;
```

```
    }
```

```
    public void setCAR_CODIGO(Integer cAR_CODIGO) {
```

```
        CAR_CODIGO = cAR_CODIGO;
```

```
    }
```

```
    public String getNOM_CARRERA() {
```

```
        return NOM_CARRERA;
```

```
    }
```

```
    public void setNOM_CARRERA(String nOM_CARRERA) {
```

```
        NOM_CARRERA = nOM_CARRERA;
```

```
    }
```

```
    public Integer getTTG_CODIGO() {
```

```
        return TTG_CODIGO;
```

```
    }
```

Se declaran los
parámetros.

```

public void setTTG_CODIGO(Integer tTG_CODIGO) {
    TTG_CODIGO = tTG_CODIGO;
}
public String getTTG_DESCRIPCION() {
    return TTG_DESCRIPCION;
}
public void setTTG_DESCRIPCION(String tTG_DESCRIPCION) {
    TTG_DESCRIPCION = tTG_DESCRIPCION;
}
public String getTTG_ANULADO() {
    return TTG_ANULADO;
}
public void setTTG_ANULADO(String tTG_ANULADO) {
    TTG_ANULADO = tTG_ANULADO;
}
public Integer getId_capitulo() {
    return id_capitulo;
}
public void setId_capitulo(Integer id_capitulo) {
    this.id_capitulo = id_capitulo;
}
public Integer getTotal() {
    return total;
}
public void setTotal(Integer total) {
    this.total = total;
}
public Integer getCODIGO() {
    return CODIGO;
}
public void setCODIGO(Integer cODIGO) {
    CODIGO = cODIGO;
}
public String getTUTOR() {
    return TUTOR;
}
public void setTUTOR(String tUTOR) {
    TUTOR = tUTOR;
}
public Integer getContador() {
    return Contador;
}
public void setContador(Integer contador) {
    Contador = contador;
}
public String getPcar_codigo() {
    return pcar_codigo;
}
public void setPcar_codigo(String pcar_codigo) {
    this.pcar_codigo = pcar_codigo;
}
public String getPcllc_rut() {
    return pcllc_rut;
}
public void setPcllc_rut(String pcllc_rut) {
    this.pcllc_rut = pcllc_rut;
}

```

Se generaron getter and setter para poder posteriormente asignarle valores para luego invocarlos.

```

    }
    public String getPcllc_nmb() {
        return pcllc_nmb;
    }
    public void setPcllc_nmb(String pcllc_nmb) {

        this.pcllc_nmb = pcllc_nmb;}

    public String getPttg_codigo_titulo() {

        return pttg_codigo_titulo;

    }

    public void setPttg_codigo_titulo(String pttg_codigo_titulo) {

        this.pttg_codigo_titulo = pttg_codigo_titulo;

    }

    public String getPttg_descripcion() {

        return pttg_descripcion;

    }

    public void setPttg_descripcion(String pttg_descripcion) {

        this.pttg_descripcion = pttg_descripcion;

    }

    public String getPDRT_FECHA_APROBACION() {

        return PDRT_FECHA_APROBACION;

    }

    public void setPDRT_FECHA_APROBACION(String pDRT_FECHA_APROBACION) {

        PDRT_FECHA_APROBACION = pDRT_FECHA_APROBACION;

    }

}

```

Entidad SMT_Avances

```
package ups.edu.entidad ;

public class SMT_Avances {

    //CAPITULOS
    private Integer id_monitoreo;
    private String capitulos;

    //SUBCAPITULOS
    private Integer ID_COMPONETE;
    private Integer ID_MONITOREO;
    private String SUB_CAPITULOS;
    private String porcentaje;
    private String fecha;

    //AVANCE
    private Integer id_avance;
    private Integer ID_SUB_CAP;
    private Integer id_tesis;
    private String ESTADO;

    public String getFecha() {
        return fecha;
    }
    public void setFecha(String fecha) {
        this.fecha = fecha;
    }

    public Integer getId_monitoreo() {
        return id_monitoreo;
    }
    public void setId_monitoreo(Integer id_monitoreo) {
        this.id_monitoreo = id_monitoreo;
    }
    public String getCapitulos() {
        return capitulos;
    }
    public void setCapitulos(String capitulos) {
        this.capitulos = capitulos;
    }
    public Integer getID_COMPONETE() {
        return ID_COMPONETE;
    }
    public void setID_COMPONETE(Integer iD_COMPONETE) {
        ID_COMPONETE = iD_COMPONETE;
    }

    public Integer getID_MONITOREO() {
        return ID_MONITOREO;
    }
}
```

```

}

public void setID_MONITOREO(Integer iD_MONITOREO) {
    ID_MONITOREO = iD_MONITOREO;
}
public String getSUB_CAPITULOS() {
    return SUB_CAPITULOS;
}
public void setSUB_CAPITULOS(String sUB_CAPITULOS) {
    SUB_CAPITULOS = sUB_CAPITULOS;
}
public String getPorcentaje() {
    return porcentaje;
}

```

```

public void setPorcentaje(String porcentaje) {

    this.porcentaje = porcentaje;

}

public Integer getId_avance() {

    return id_avance;

}

public void setId_avance(Integer id_avance) {

    this.id_avance = id_avance;

}

public Integer getID_SUB_CAP() {

    return ID_SUB_CAP;

}

public void setID_SUB_CAP(Integer iD_SUB_CAP) {

    ID_SUB_CAP = iD_SUB_CAP;

}

public Integer getId_tesis() {

    return id_tesis;

}

public void setId_tesis(Integer id_tesis) {

    this.id_tesis = id_tesis;

}

```



```

    }

    public String getESTADO() {

        return ESTADO;

    }

    public void setESTADO(String eESTADO) {

        ESTADO = eESTADO;

    }

}

```

Entidad SMT_Menues

```

package ups.edu.entidad;

public class SMT_Menues {

    //MENU
    private String descripcion;
    private String pagina;
    private String target;

    //SUBMENU
    private String descripciones;
    private String paginas;
    private String targets;
    private String ids;

    public String getDescripcion() {
        return descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    public String getPagina() {
        return pagina;
    }
    public void setPagina(String pagina) {
        this.pagina = pagina;
    }
    public String getTarget() {
        return target;
    }
    public void setTarget(String target) {
        this.target = target;
    }
    public String getDescripciones() {
        return descripciones;
    }
}

```

```

    }
    public void setDescripciones(String descripciones) {
        this.descripciones = descripciones;
    }
    public String getPaginas() {
        return paginas;
    }
    public void setPaginas(String paginas) {
        this.paginas = paginas;
    }
    public String getTargets() {
        return targets;
    }
    public void setTargets(String targets) {
        this.targets = targets;
    }
    public String getIds() {
        return ids;
    }
    public void setIds(String ids) {
        this.ids = ids;
    }
}

```

Entidad SMT_Usuarios

```

package ups.edu.entidad;

public class SMT_Usuario {

    public Integer Idusuario;

    public String Usuario;

    public String Clave;

    public Integer Admin;

    public Integer getIdusuario() {
        return Idusuario;
    }
    public void setIdusuario(Integer idusuario) {
        Idusuario = idusuario;
    }
    public String getUsuario() {
        return Usuario;
    }
    public void setUsuario(String usuario) {
        Usuario = usuario;
    }
    public String getClave() {
        return Clave;
    }
}

```

```

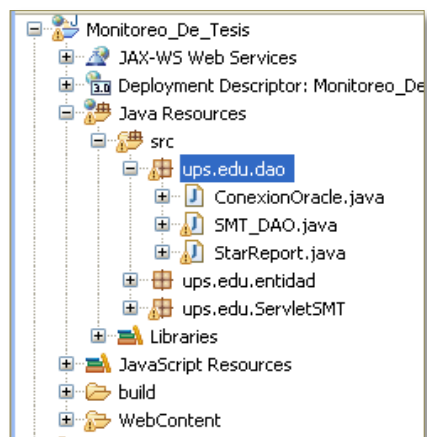
    public void setClave(String clave) {
        Clave = clave;
    }
    public Integer getAdmin() {
        return Admin;
    }
    public void setAdmin(Integer admin) {
        Admin = admin;
    }
}

```

3.3.3 Capa del Controlador

Esta capa contiene las clases asociadas a las entidades de la base de datos, las cuales hacen referencia a los procedimientos y los diferentes parámetros creados en la capa de entidad, además contiene la clase conexión.

La capa controlador permitirá establecer la conexión entre la capa de presentación (Vista), para recibir las solicitudes y presentar los resultados, y la capa de acceso a datos (Modelo) permitirá solicitar a la base de datos almacenar o recuperar datos.



Se creó una clase para establecer la conexión a base de datos con el sistema, creando varios métodos como para inicializar la conexión el que contiene los parámetros necesarios para iniciarla, y otros como cerrar, y hacer commit

Clase SMT_DAO

Clase pública que permite realizar procedimientos utilizando las entidades identificando cada parámetro de entrada inicializados en la clase entidad.

Se creó métodos públicos string y list que hace referencia a la clase entidad, en estos se forma la sentencia sql que va a realizar la llamada al procedimiento contenido dentro de un paquete en la base de datos o ya sea solo para realizar una consulta.

Se crea un objeto callablestatement estableciendo la conexión y realizando el llamado a la sentencia sql cuyo valor inicial era nulo, se crea una variable la result en la cual se almacenara el resultado de la ejecución cuyo valor inicial era nulo.

Se crea un manejador de error el cual contendrá el procedimiento y si al ejecutase se obtiene un resultado en una variable de salida enviada por el procedimiento ya sea “true” si se realizó exitoso o “false” si ocurrió algún error.

Se envía un mensaje visible para el usuario de los resultados ejecutados por el procedimiento en el caso de que así sea, a continuación vemos cada método:

Métodos de Conexión a la base

Se creó métodos para establecer la conexión a base de datos con el sistema, creando varios métodos como para inicializar la conexión el que contiene los parámetros necesarios para iniciarla, y otros como cerrar, y hacer commit.

Estamos utilizando pool de conexiones el cual nos va a permitir mantener varias sesiones, así como una mejor utilización de las entidades.

```
package ups.edu.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.sql.DataSource;

import ups.edu.entidad.SMT_Tesis;

import ups.edu.entidad.SMT_Usuario;

import ups.edu.entidad.SMT_Menues;

import ups.edu.entidad.SMT_Avances;

//import ups.edu.dao.ConexionOracle;

import java.io.FileOutputStream;

import java.io.FileInputStream;

import java.io.IOException;

import net.sf.jasperreports.engine.JRException;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.util.JRLoader;

import net.sf.jasperreports.view.JasperViewer;

import org.apache.commons.net.ftp.FTPClient;
```

```

public class SMT_DAO {

    Connection connection;

    private boolean connectionFree = true;

    private ArrayList usuarios;

    private ArrayList data;

    public SMT_DAO() throws Exception {

        try {

            Context initContext = new InitialContext();

            Context envContext = (Context) initContext.lookup("java:/comp/env/");

            DataSource ds = (DataSource) envContext.lookup("jdbc-mysql2");

            this.connection = ds.getConnection();

        } catch (Exception e) {
            throw new Exception(
                "No se pudo abrir la base de datos biblioteca: "+
e.getMessage());
        }
    }

    public Connection getCon() {

        return this.connection;

    }

    protected synchronized Connection getConnection() {
        while (this.connectionFree == false) {
            try {
                wait();
            } catch (InterruptedException e) {

            }

        }
        this.connectionFree = false;
        notify();
        return this.connection;
    }

    public synchronized void releaseConnection() {
        while (this.connectionFree == true) {
            try {
                wait();
            } catch (InterruptedException e) {

            }
        }
    }
}

```

```

    }
    this.connectionFree = true;
    notify();
}

protected synchronized void CloseConnection() throws Exception {

    this.connection.close();

}

```

Método para validar inicio de sesión

Es este se reciben los parámetros de nombre de usuario, clave, y el valor le enviamos 1 desde la capa de aplicación este valor es para cuando se desea que solo inicien una sola sesión de usuario, en este primero se valida si el nombre del usuario existe en la tabla, luego valida si ya tiene una sesión activa y de tenerla no le permite ingresar, y si no tiene sesión activa pasa a verificar la clave desencriptándola ya que en la base se encuentra encriptado de ser la correcta permite ingresar al sistema.

```

public String ValidaUser(String user, String pass, String valor) {
    String query = null;
    try {
        this.getConnection();
        query = "select * from mtesis.usuarios where upper(usuario) = '"
            + user.toUpperCase() + "'";
        String password = " ";
        String sesion = " ";
        String estado = " ";
        PreparedStatement preparedStatement =
this.connection.prepareStatement(query);
        ResultSet resultSet = preparedStatement.executeQuery();
        System.out.println("1: " + password);
        if (resultSet.next()) {
            password = resultSet.getString(2);
            sesion = resultSet.getString(6);
            estado = resultSet.getString(5);
            System.out.println("2: " + password);
            if (estado.equals("A")) {

                String clave = this.Desencriptar(password);
                System.out.println("3: " + clave);

                if (clave.equals(pass)) {

                    if (valor.equals("0")) {

                        if (sesion.equals("I"))
                            return "true";
                        else

```

```

        return "Ya tiene abierta otra sesión";
    } else
        return "true";
    } // fin password
    else
        return "Password Incorrecto";
    } // fin estado A
    Else
        return "Usuario está inactivo";
    // fin else pass
    } // fin resultset

    else
        return "Usuario no existe";
        // fin else estado A
    } // fin try
    catch (SQLException e) {
        this.releaseConnection ();
        return e.toString() + "-" + query;
    }
}

```

Método para activar sesión

En este método se cambia el estado de sesión del usuario para que no vuelva a iniciar sesión dependiendo de la validación en el método que permite ingresar al sistema.

```

public String ActivaSesion(String user) {

    try {
        int rows = 0;
        this.getConnection();
        String query = "update mtesis.usuarios set sesion='A' where upper(usuario) = '"
            + user.toUpperCase() + "'";
        PreparedStatement preparedStatement =
        this.connection.prepareStatement(query);
        rows = preparedStatement.executeUpdate();
        return "true";

    } catch (SQLException e) {
        this.releaseConnection ();
        return e.toString();
    }
}

```


Método para inactivar sesión

En este método se cambia el estado de sesión del usuario para que no vuelva a iniciar sesión dependiendo de la validación en el método que permite ingresar al sistema.

```
public String InactivaSesion(String user) {  
  
    try {  
        int rows = 0;  
        this.getConnection();  
        String query = "update mtesis.usuarios set sesion='I' where upper(usuario) = '"  
            + user.toUpperCase() + "'";  
        PreparedStatement preparedStatement =  
this.connection.prepareStatement(query);  
        rows = preparedStatement.executeUpdate();  
        return "true";  
    } catch (SQLException e) {  
        this.releaseConnection();  
        return e.toString();  
    }  
}
```

Método Encriptar y Desencriptar

Estos métodos se utilizan para encriptar la clave al realizar el ingreso a la base y desencriptarla al momento de compararla en el inicio de sesión de usuario.

```
public String Encriptar(String pass) {  
  
    char letra_as = ' ';  
    String palabra_as = "";  
    int num_ascii = 0;  
    String str_num_ascii = "";  
    Integer numero = null;  
    int len_ascii = 0;  
  
    for (int ind = 0; ind < pass.length(); ind++) {  
        letra_as = pass.charAt(ind);  
  
        num_ascii = (int) letra_as;  
        numero = new Integer(num_ascii);  
        if (numero.toString().length() == 1) {  
            palabra_as = palabra_as + "00" + str_num_ascii;  
        } else {  
            if (numero.toString().length() == 2) {  
                palabra_as = palabra_as + "0" + numero.toString();  
            } else {  
                palabra_as = palabra_as + numero.toString();  
            }  
        }  
    }  
}
```

```

        }
    }
    String impares = "";
    String pares = "";
    for (int ind = 0; ind < palabra_as.length(); ind += 2) {
        impares = impares + palabra_as.charAt(ind);
    }
    for (int ind = 1; ind < palabra_as.length(); ind += 2) {
        pares = pares + palabra_as.charAt(ind);
    }
    String encriptada = impares + pares;

    return encriptada;
}
/*****DESENCRIPTACLAVE*****/

public String Desencriptar(String pass) {

    int len_encriptada = pass.length();
    int len_impares = Math.round(len_encriptada / (float) 2);
    int len_pares = len_encriptada - len_impares;
    String impares_d = pass.substring(0, len_impares);
    String pares_d = pass.substring(len_impares, len_encriptada);
    String desencriptada = "";
    int control = 0;
    if (len_impares > len_pares)
        control = 1;
    for (int ind = 0; ind < len_impares; ind++) {
        desencriptada = desencriptada + pass.charAt(ind);
        if (ind < len_impares - control)
            desencriptada = desencriptada
ind                                     + pass.substring(len_impares + ind, len_impares +
                                     + 1);
    }
    String desencriptada_fin = "";
    for (int ind = 0; ind < desencriptada.length(); ind += 3) {
        int ascii = Integer.parseInt(desencriptada.substring(ind, ind + 3));
        char asc = (char) ascii;
        desencriptada_fin = desencriptada_fin + asc;
    }
    return desencriptada_fin;
}

```

Método para generar menú principal

En este método se obtiene las opciones del menú principal dependiendo del tipo de usuario en cual lo recibe como parámetro de entrada.

```

public List<SMT_Menues> GeneraMenu1(String user) throws Exception

this.getConnection ();
String sql = "select * from mtesis.smt_perfil p, mtesis.smt_perfil_x_opcion_menu po, mtesis.usuarios u,
mtesis.mt_opcion_menu m ";
sql = sql + " where upper(u.usuario) = '" + user.toUpperCase() + "' ";
sql = sql+ " and po.perfil = u.Perfil and p.id_perfil = po.perfil and p.estado = 'A' and m.id_opcion=
po.opcion and m.op_padre=0";
ResultSet rs = null; // trae los registro de la consulta
PreparedStatement pst;
List<SMT_Menues> menus = null;
menus = new ArrayList<SMT_Menues>();

    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        rs = pst.executeQuery();
        if (rs != null) { // valida q tenga dat1os la consulta
            while (rs.next()) {
                SMT_Menues be = new SMT_Menues();
                be.setDescripcion(rs.getString(13));
                be.setPagina(rs.getString(12));
                be.setTarget(rs.getString(17));
                menus.add(be);
            }
        }
    } catch (SQLException ex) {
        this.releaseConnection ();
        ex.printStackTrace();
    } finally {
    }
    return menus;
}

```

Método para generar submenú

En este método se obtiene las submenús dependiendo del tipo de usuario en cual lo recibe como parámetro de entrada y además su opción principal a la cual pertenece.

```

public List<SMT_Menues> GeneraSubMenu1(String op_padre, String user) throws Exception {
    String sql = "select * from mtesis.mt_opcion_menu ";
    sql = sql + " where op_padre = " + op_padre;
    sql = sql+ " and id_opcion in (select po.opcion from mtesis.smt_perfil p,
mtesis.smt_perfil_x_opcion_menu po, mtesis.usuarios u ";
    sql = sql + " where upper(u.usuario) = '" + user.toUpperCase() + "'";
    sql = sql+ " and po.perfil = u.perfil and p.id_perfil = po.perfil and p.estado = 'A') and
estado = 'A' order by descripcion ";

    ResultSet rs = null; // trae los registro de la consulta
    PreparedStatement pst;
    List<SMT_Menues> submenus = null;
    submenus = new ArrayList<SMT_Menues>();
}

```

```

    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        rs = pst.executeQuery();
        if (rs != null) { // valida q tenga dat1os la consulta
            while (rs.next()) {
                SMT_Menues be = new SMT_Menues();
                be.setIds(rs.getString(2));
                be.setDescripciones(rs.getString(2));
                be.setPaginas(rs.getString(3));
                be.setTargets(rs.getString(6));

                System.out.println(rs.getString(2));
                submenues.add(be); // para no perder los datos c/v q se
            }
        }
    } catch (SQLException ex) {
        this.releaseConnection ();
        ex.printStackTrace();
    } finally {
    }
    return submenues;
}

```

Método para cambiar la clave de usuario

En este método se invoca al procedimiento de la base de datos que permite actualizar la clave del usuario, recibiendo como parámetro solo el usuario y la clave.

```

public String CambiaPassword(String user, String pass) throws SQLException {

    String call = "";
    java.sql.CallableStatement cl = null;
    this.getConnection ();
    // boolean resultado = false;
    try {
        String paquete = "{ call
SMTK_USUARIOS.SEK_ACTUALIZAR_CTUTOR(?,?,?)}";
        cl = this.connection.prepareCall(paquete);
        cl.setString(1, user.toString());
        cl.setString(2, pass.toString());
        cl.registerOutParameter(3, java.sql.Types.VARCHAR);
        cl.execute();
        call = cl.getString(3);
        return "true";
    } catch (Exception ex) {
        //realizarRollback();
    }
}

```

```

        ex.printStackTrace();
        return ex.toString();
    } finally {
        cl.close();
    }
}

```

Métodos de Consulta

Varios métodos de consulta están presentes en esta clase los cuales me va a devolver todos los datos necesarios en las entidades.

Consulta_carrera()

```

Public List<SMT_Tesis> consulta_carrera() throws Exception {
    String query = null;
    this.getConnection();

    String sql = "select c.cai_codigo, upper(c.cai_descripcion) from SNA.sna_carrera_institucion c order by c.cai_codigo";

    ResultSet rs = null;
    PreparedStatement pst;
    List<SMT_Tesis> carrera = null;
    carrera = new ArrayList<SMT_Tesis>();
    try {
        pst = this.connection.prepareStatement(sql);
        rs = pst.executeQuery();
        if (rs != null) {
            while (rs.next()) {
                SMT_Tesis car = new SMT_Tesis();
                car.setCAR_CODIGO(rs.getInt(1));
                car.setNOM_CARRERA(rs.getString(2));
                carrera.add(car);
            }
        }
    } catch (SQLException ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally {
    }
    return carrera;
}

```

Consulta_tesis(String idcar, String idtut)

```
Public List<SMT_Tesis> consulta_tesis(String idcar, String idtut) throws Exception {
    this.getConnection();

    String sql = "select distinct r.car_codigo,u.usuario, sr.codigo,tm.ttg_codigo,upper(TM.ttg_descripcion) "
        +"from sna.sna_det_res_tem_tes t, ghadmin.rh_person rh,sna.sna_titulo_trabajo_grado TM,"
        +"sna.sna_resolucion_trabajo_grado r,sna.sna_resolucion_tema_tesis sr, mtesis.usuarios u,"
        +" mtesis.mt_periodos mp, mtesis.mt_plazos mtpla where T.TTG_CODIGO_TITULO = "
        +" TM.TTG_CODIGO " +
        +"and r.car_codigo = ? and upper(u.usuario) = upper(?) and sr.codigo = u.id_usuario AND "
        +"rh.codigo = sr.codigo and sr.ttg_codigo_titulo = tm.ttg_codigo and t.drt_codigo = "
        +" r.drt_codigo_apr " +
        +"and mp.id_tran = 1 and mtpla.descripcion='prorroga' " +
        +"and add_months(t.drt_fecha_aprobacion, " +
        +"(mp.duracion_meses*(mtpla.num_plazos+1))) > sysdate and r.sed_codigo = 3 and "
        +"r.rtg_eliminado = 'N'" + " order by upper(TM.ttg_descripcion) asc";

    ResultSet rs = null; // trae los registro de la consulta
    PreparedStatement pst;
    List<SMT_Tesis> tesis = null;
    tesis = new ArrayList<SMT_Tesis>();
    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        pst.setString(1, idcar);
        pst.setString(2, idtut);
        rs = pst.executeQuery();
        if (rs != null) { // valida q tenga dat1os la consulta
            while (rs.next()) {
                SMT_Tesis t = new SMT_Tesis();
                t.setTTG_CODIGO(rs.getInt(4));
                t.setTTG_DESCRIPCION(rs.getString(5));
                tesis.add(t); // para no perder los daros c/v q se recorre se
                                // agrega a una lista
            }
        }
    } catch (SQLException ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally
    {
        return tesis;
    }
}
```

Autenticar (String user)

```
Public List<SMT_Usuario> autenticar(String user) throws SQLException {
    this.getConnection();

    String sql = "select upper(usuario), id_usuario from mtesis.usuarios where upper(usuario)= upper(?)";
    PreparedStatement ps = null;// prepara la sentencia
    ResultSet rs = null;// trae los registros
    List<SMT_Usuario> usua = null;

    try {
        ps = this.connection.prepareStatement(sql);// envio el sql le asigno a con
        ps.setString(1, user);
        rs = ps.executeQuery();// ejecuto el sql
        usua = new ArrayList<SMT_Usuario>();
        while (rs.next()) {
            SMT_Usuario usu = new SMT_Usuario();
            usu.setUsuario(rs.getString(1));
            usu.setIdusuario(rs.getInt(2));
            usua.add(usu);
        }
    } catch (Exception ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally
    {
    }
    return usua;
}
```

Consulta_tutor_ing(String idcar)

```
Public List<SMT_Tesis> consulta_tutor_ing(String idcar) throws Exception {
    this.getConnection();

    String sql = "select distinct r.car_codigo, rh.codigo, (rh.apellido || ' ' || rh.nombre) " +
        "from sna.sna_det_res_tem_tes t, ghadmin.rh_person rh," +
        "sna.sna_titulo_trabajo_grado TM, sna.sna_resolucion_trabajo_grado r," +
        "sna.sna_resolucion_tema_tesis sr, mtesis.usuarios u, mtesis.mt_periodos mp," +
        "mt_plazos mtpla where T.TTG_CODIGO_TITULO = TM.TTG_CODIGO " +
        "AND rh.codigo = sr.codigo and sr.ttg_codigo_titulo = tm.ttg_codigo " +
        "and t.drt_codigo = r.drt_codigo_apr and r.car_codigo = ? " +
        "and r.sed_codigo = 3 and r.rtg_eliminado = 'N' " +
        "AND t.drt_fecha_aprobacion between to_date('03/03/2012', 'dd/mm/yyyy') " +
        "and sysdate and mp.id_tran = 1 and add_months(t.drt_fecha_aprobacion, " +
        "(mp.duracion_meses*(mtpla.num_plazos+1))) > sysdate" +
        " and mtpla.descripcion='prorroga' order by (rh.apellido || ' ' || rh.nombre) asc";

    ResultSet rs = null; // trae los registro de la consulta
    PreparedStatement pst;
    List<SMT_Tesis> tutorIng = null;
    tutorIng = new ArrayList<SMT_Tesis>();
    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        pst.setString(1, idcar);
        rs = pst.executeQuery();
        if (rs != null) { // valida q tenga dat1os la consulta
            while (rs.next()) {
                SMT_Tesis te = new SMT_Tesis();
                System.out.println(idcar);
                te.setCODIGO(rs.getInt(2));
                te.setTUTOR(rs.getString(3));
                tutorIng.add(te); // para no perder los daros c/v q se
            }
        }
    } catch (SQLException ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally
    {
    }
    return tutorIng;
}
```


consulta_tutor_man(String idcar)

```
Public List<SMT_Tesis> consulta_tutor_man(String idcar, String usu) throws Exception {
    this.getConnection();

    String sql = "select distinct r.car_codigo, rh.codigo, " +
        "(rh.apellido || ' ' || rh.nombre) from sna.sna_det_res_tem_tes t," +
        " ghadmin.rh_person rh, sna.sna_titulo_trabajo_grado TM," +
        " sna.sna_resolucion_trabajo_grado r, sna.sna_resolucion_tema_tesis sr," +
        " mtesis.usuarios u, mtesis.mt_periodos mp, mt_plazos mtpla " +
        "where T.TTG_CODIGO_TITULO = TM.TTG_CODIGO " +
        "AND rh.codigo = sr.codigo and sr.ttg_codigo_titulo = tm.ttg_codigo" +
        " and t.drt_codigo = r.drt_codigo_apr and r.car_codigo = ? " +
        "and r.sed_codigo = 3 and r.rtg_eliminado = 'N' " +
        "AND t.drt_fecha_aprobacion between to_date('03/03/2012', 'dd/mm/yyyy')" +
        " and sysdate and mp.id_tran = 1 and " +
        "add_months(t.drt_fecha_aprobacion, (mp.duracion_meses*(mtpla.num_plazos+1))) > sysdate " +
        "+and u.id_usuario = "+usu+" and u.id_usuario = rh.codigo and u.estado='A' " +
        "and mtpla.descripcion='prorroga' " +
        "order by (rh.apellido || ' ' || rh.nombre) asc";

    ResultSet rs = null; // trae los registro de la consulta
    PreparedStatement pst;
    List<SMT_Tesis> tutorman = null;
    tutorman = new ArrayList<SMT_Tesis>();

    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        pst.setString(1, idcar);
        rs = pst.executeQuery();
        if (rs != null) { // valida q tenga dat1os la consulta
            while (rs.next()) {

                SMT_Tesis tm = new SMT_Tesis();
                System.out.println(idcar);
                tm.setCODIGO(rs.getInt(2));
                tm.setTUTOR(rs.getString(3));
                tutorman.add(tm); // para no perder los daros c/v q se

            }
        }
    } catch (SQLException ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally {
    }
    return tutorman;
}
```

buscar_alumno(String idcar, String idalum)

```
Public List<SMT_Tesis> buscar_alumno(String idcar, String idalum) throws Exception {
    this.getConnection();

    String sql = "select distinct t.cllic_rut, r.car_codigo, t.cllic_nmb ," +
        " titulo.ttg_codigo_titulo, titulo.ttg_descripcion, t.cllic_cdg ," +
        " to_char(mtp.fecha_inicio_pro,'yyyy-mm-dd'), " +
        "to_char(mtp.fecha_fin_pro,'yyyy-mm-dd'), mtp.status_tes, " +
        "to_char(ADD_MONTHS(mtp.fecha_fin_pro,mtp.duracion_meses),'yyyy-mm-dd')," +
        " to_char(dt.drt_fecha_culmina_derecho,'yyyy-mm-dd'), mtp.num_pro " +
        "from (select dt.ttg_codigo_titulo, tg.ttg_descripcion " +
        "from sigac.cliente_local t, sna.sna_det_res_tem_tes dt, " +
        "sna.sna_titulo_trabajo_grado tg where trim(t.cllic_rut)=trim(?) " +
        "and trim(dt.cllic_cdg) = trim(t.cllic_cdg) " +
        "and trim(tg.ttg_codigo)= trim(dt.ttg_codigo_titulo) " +
        "and tg.ttg_anulado='N') titulo, sna.sna_det_res_tem_tes dt, " +
        "sigac.cliente_local t, sna.sna_resolucion_trabajo_grado r," +
        " mtesis.mt_prorrogas mtp, mtesis.mt_periodos mtpe " +
        "where trim(dt.ttg_codigo_titulo) = trim(titulo.ttg_codigo_titulo)" +
        " and trim(t.cllic_cdg) = trim(dt.cllic_cdg) " +
        "and trim(r.drt_codigo_apr) = trim(dt.drt_codigo) " +
        "and trim(r.car_codigo) = trim(?) and trim(r.sed_codigo) = 3 " +
        "and trim(r.rtg_eliminado) = 'N' and mtp.id_tesis = titulo.ttg_codigo_titulo " +
        "and mtpe.nombre_tran='TESIS' order by mtp.num_pro desc";

    ResultSet rs = null; // trae los registro de la consulta
    PreparedStatement pst;
    List<SMT_Tesis> alumno = null;
    alumno = new ArrayList<SMT_Tesis>();
    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        pst.setString(1, idalum);
        pst.setString(2, idcar);
        rs = pst.executeQuery();
        while (rs.next()) {
            SMT_Tesis ba = new SMT_Tesis();
            ba.setPcllc_rut(rs.getString(1));
            ba.setPcllc_nmb(rs.getString(3));
            ba.setPttg_codigo_titulo(rs.getString(4));
            ba.setPttg_descripcion(rs.getString(5));
            ba.setPcllc_codigo(rs.getString(6));
            ba.setMtp_fi(rs.getString(7));
            ba.setMtp_ff(rs.getString(8));
            ba.setMtp_estado(rs.getString(9));
            ba.setMtp_proff(rs.getString(10));
            ba.setMtp_c1d(rs.getString(11));
            ba.setMtp_num_pro(rs.getInt(12));
            alumno.add(ba);
        }
    } catch (SQLException ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally {}
    return alumno;
}
```

buscarcapitulo()

```
Public List<SMT_Avances> buscarcapitulo() throws Exception {
    this.getConnection();

    String sql = "select ID_MONITOREO, upper(capitulos) from
                  mtesis.MT_CAPITULOS";

    ResultSet rs = null; // trae los registro de la consulta
    PreparedStatement pst;
    List<SMT_Avances> cap = null;
    cap = new ArrayList<SMT_Avances>();
    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        rs = pst.executeQuery();
        if (rs != null) { // valida q tenga dat1os la consulta
            while (rs.next()) {
                SMT_Avances bc = new SMT_Avances();
                bc.setID_MONITOREO(rs.getInt(1));
                bc.setCapitulos(rs.getString(2));
                cap.add(bc); // para no perder los daros c/v q se recorre se
            }
        }
    } catch (SQLException ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally
    {
    }
    return cap;
}
```

buscarsubcapitulo(String idc)

```
Public List<SMT_Avances> buscarsubcapitulo(String idc) throws Exception {
    this.getConnection();

    String sql = "select ID_MONITOREO, ID_COMPONETE, SUB_CAPITULOS " +
        "from mtesis.MT_SUB_CAPITULOS where trim(ID_MONITOREO) = (?)";

    ResultSet rs = null; // trae los registro de la consulta
    PreparedStatement pst;
    List<SMT_Avances> subcap = null;
    subcap = new ArrayList<SMT_Avances>();
    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        pst.setString(1, idc);
        rs = pst.executeQuery();
        if (rs != null) { // valida q tenga dat1os la consulta
            while (rs.next()) {
                SMT_Avances bsc = new SMT_Avances();
                bsc.setID_COMPONETE(rs.getInt(2));
                bsc.setSUB_CAPITULOS(rs.getString(3));
                subcap.add(bsc); // para no perder los daros c/v q se recorre
            }
        }
    } catch (SQLException ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally
    {
    }
    return subcap;
}
```

buscarsubcapituloM(String idt, String cp)

```
Public List<SMT_Avances> buscarsubcapituloM(String idt, String cp) throws Exception {
    this.getConnection();

    String sql = "select  c.id_monitoreo, a.id_tesis, c.capitulos, a.id_sub_cap, " +
        "upper(sc.sub_capitulos), upper(a.id_estado), a.fecha from mtesis.MT_AVANCES a," +
        " mtesis.mt_sub_capitulos sc, mtesis.mt_capitulos c " +
        "where trim(a.id_tesis) = trim(?) and trim(c.id_monitoreo) = trim(?) " +
        "and trim(sc.id_componete) = trim(a.id_sub_cap) and " +
        "trim(sc.id_monitoreo) = trim(c.id_monitoreo)";

    ResultSet rs = null; // trae los registro de la consulta
    PreparedStatement pst;
    List<SMT_Avances> subcapm = null;
    subcapm = new ArrayList<SMT_Avances>();
    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        pst.setString(1, idt);
        pst.setString(2, cp);
        rs = pst.executeQuery();
        if (rs != null) { // valida q tenga dat1os la consulta
            while (rs.next()) {
                SMT_Avances bscm = new SMT_Avances();
                bscm.setCapitulos(rs.getString(3));
                bscm.setID_COMPONETE(rs.getInt(4));
                bscm.setSUB_CAPITULOS(rs.getString(5));
                bscm.setESTADO(rs.getString(6));
                bscm.setFecha(rs.getString(7));
                subcapm.add(bscm); // para no perder los daros c/v q se
            }
        }
    } catch (SQLException ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally
    {
    }
    return subcapm;
}
}
```

verificar_monitoreo(String idt)

```
Public List<SMT_Tesis> verificar_monitoreo(String idt) throws Exception {
    this.getConnection();

    String sql = "select a.id_tesis, count(a.id_avance) " +
        "from mtesis.mt_avances a where trim(a.id_tesis) = trim(?) group by a.id_tesis";

    ResultSet rs = null; // trae los registro de la consulta
    PreparedStatement pst;
    List<SMT_Tesis> verMon = null;
    verMon = new ArrayList<SMT_Tesis>();
    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        pst.setString(1, idt);
        rs = pst.executeQuery();
        if (rs != null) { // valida q tenga dat1os la consulta
            while (rs.next()) {
                SMT_Tesis vm = new SMT_Tesis();
                vm.setContador(rs.getInt(2));
                verMon.add(vm); // para no perder los daros c/v q se recorre
            }
        }
    } catch (SQLException ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally {
    }
    return verMon;
}
```

capturar_usuario(String user)

```
Public List<SMT_Usuario> capturar_usuario(String user) throws SQLException {
    this.getConnection();

    String sql = "select rh.codigo,substr(rh.nombre,1,1)||" +
        "substr(rh.apellido,1,instr(rh.apellido,' ')-1)" +
        "||substr(rh.apellido,instr(rh.apellido,'')+1,1)" +
        "from ghadmin.rh_person rh where rh.codigo=?";

    PreparedStatement ps = null;// prepara la sentencia
    ResultSet rs = null;// trae los registros
    List<SMT_Usuario> capusuario = null;
    try {
        ps = this.connection.prepareStatement(sql);// envio el sql le asigno a con
        ps.setString(1, user);
        rs = ps.executeQuery();// ejecuto el sql
        capusuario = new ArrayList<SMT_Usuario>();
        while (rs.next()) {
            SMT_Usuario cu = new SMT_Usuario();
            cu.setIdusuario(rs.getInt(1));
            cu.setUsuario(rs.getString(2));
            capusuario.add(cu);
        }
    } catch (Exception ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally {
        if (rs != null)
            rs.close();
        if (ps != null)
            ps.close();
    }
    return capusuario;
}
```

validar_fecha(String idt)

```
Public List<SMT_Tesis> validar_fecha(String idt) throws Exception {
    this.getConnection();

    String sql = "select distinct t.DRT_FECHA_APROBACION from sna.SNA_DET_RES_TEM_TES t," +
        " mtesis.mt_periodos mp, mt_plazos mtpla where t.TTG_CODIGO_TITULO = ? " +
        "and mp.id_tran = 1 and " +
        "months_between(sysdate, t.drt_fecha_aprobacion) >= " +
        "(mp.duracion_meses*(mtpla.num_plazos+1)) " +
        "and mtpla.descripcion='prorroga'";

    ResultSet rs = null; // trae los registro de la consulta
    PreparedStatement pst;
    List<SMT_Tesis> lBecario = null;
    lBecario = new ArrayList<SMT_Tesis>();
    try {
        pst = this.connection.prepareStatement(sql); // preparar pa la conexion
        pst.setString(1, idt);
        rs = pst.executeQuery();
        if (rs != null) { // valida q tenga dat1os la consulta
            while (rs.next()) {
                SMT_Tesis be = new SMT_Tesis();
                be.setPDRT_FECHA_APROBACION(rs.getString(1));
                lBecario.add(be); // para no perder los daros c/v q se
            }
        }
    } catch (SQLException ex) {
        this.releaseConnection();
        ex.printStackTrace();
    } finally {
    }
    return lBecario;
}
```


Métodos de Ingreso

Varios métodos de ingreso están presentes en esta clase los cuales van a registrar los parámetros indicados a la base de datos.

Ingresar(String isc, String it, String ie, String u)

```
public String Ingresar(String isc, String it, String ie, String u) throws SQLException {
    this.getConnection();
    String call = ""; java.sql.CallableStatement cl = null;
        try {
            String paquete = "{ call SEK_TRX_SMT_AVANCES.SEK_INGRESAR(?,?,?,?)}";
                cl = this.connection.prepareCall(paquete);
                cl.setString(1, isc.toString());
                cl.setString(2, it.toString());
                cl.setString(3, ie.toString());
                cl.setString(4, u.toString());
                cl.registerOutParameter(5, java.sql.Types.VARCHAR);
                cl.execute();
                call = cl.getString(5);
                return "true";
            } catch (Exception ex) {
                this.releaseConnection();
                ex.printStackTrace();
                return "false";
            } finally {
                cl.close();
            }
        }
    }
```

IngresarUsu (String it, String p, String nt)

```
public String IngresarUsu(String it, String p, String nt) throws SQLException {
    this.getConnection();
    String call = ""; java.sql.CallableStatement cl = null;
        try {
            String paquete = "{ call SMTK_USUARIOS.SEK_INGRESAR_USUARIO(?,?,?,?)}";
                cl = this.connection.prepareCall(paquete);
                cl.setString(1, it.toString());
                cl.setString(2, p.toString());
                cl.setString(3, nt.toString());
                cl.registerOutParameter(4, java.sql.Types.VARCHAR);
                cl.execute();
                call = cl.getString(4);
                return "true";
            } catch (Exception ex) {
                this.releaseConnection(); ex.printStackTrace();
                return "false";
            } finally {
                cl.close();
            }
        }
    }
```

Ingresar_prorroga(String idt, String ida)

```
public String Ingresar_prorroga(String idt, String ida) throws SQLException {
    this.getConnection();
    String call = "";
    java.sql.CallableStatement cl = null;
        try {
            String paquete = "{ call SMTK_PRORROGA.GCP_INVOCA(?,?,?)}";
            cl = this.connection.prepareCall(paquete);
            cl.setString(1, idt.toString());
            cl.setString(2, ida.toString());
            cl.registerOutParameter(3, java.sql.Types.VARCHAR);
            cl.execute();
            call = cl.getString(3);
        return "true";
        } catch (Exception ex) {
            this.releaseConnection();
            ex.printStackTrace();
            return ex.toString();
        } finally {
            cl.close();
        }
    }
}
```

Métodos de Actualización

Varios métodos de actualización están presentes en esta clase los cuales van a permitir modificar los parámetros indicados a la base de datos.

Modificar (String isc, String it, String ie, String u)

```
public String Modificar(String isc, String it, String ie, String u) throws SQLException {
    this.getConnection();

    String call = "";
    java.sql.CallableStatement cl = null;
        try {
            String paquete = "{ call SEK_OBJ_SMT_AVANCES.SEK_ACTUALIZAR(?,?,?,?)}";
            cl = this.connection.prepareCall(paquete);
            cl.setString(1, isc.toString());
            cl.setString(2, it.toString());
            cl.setString(3, ie.toString());
            cl.setString(4, u.toString());
            cl.registerOutParameter(5, java.sql.Types.VARCHAR);
            cl.execute();
            call = cl.getString(5);
            return "true";
        } catch (Exception ex) {
            this.releaseConnection();
        }
    }
}
```

```
        ex.printStackTrace();
        return ex.toString();
    } finally {
        cl.close();
    }
}
```

Métodos de Generación de Reportes

Para invocar los reportes del sistemas.

ejecutarReporte(String template,Map param)

```
public void ejecutarReporte(String template,Map param){

    this.getConnection();
    try{

        System.out.println("Cargando reporte...");

        try{
            reporte=(JasperReport) JRLoader.loadObject(template);
        }catch(JRException e){
            e.printStackTrace();
        }
        return;
    }

    JasperPrint jasperprint= JasperFillManager.fillReport(reporte,param,this.connection);

    JasperViewer JV= new JasperViewer(jasperprint,false);

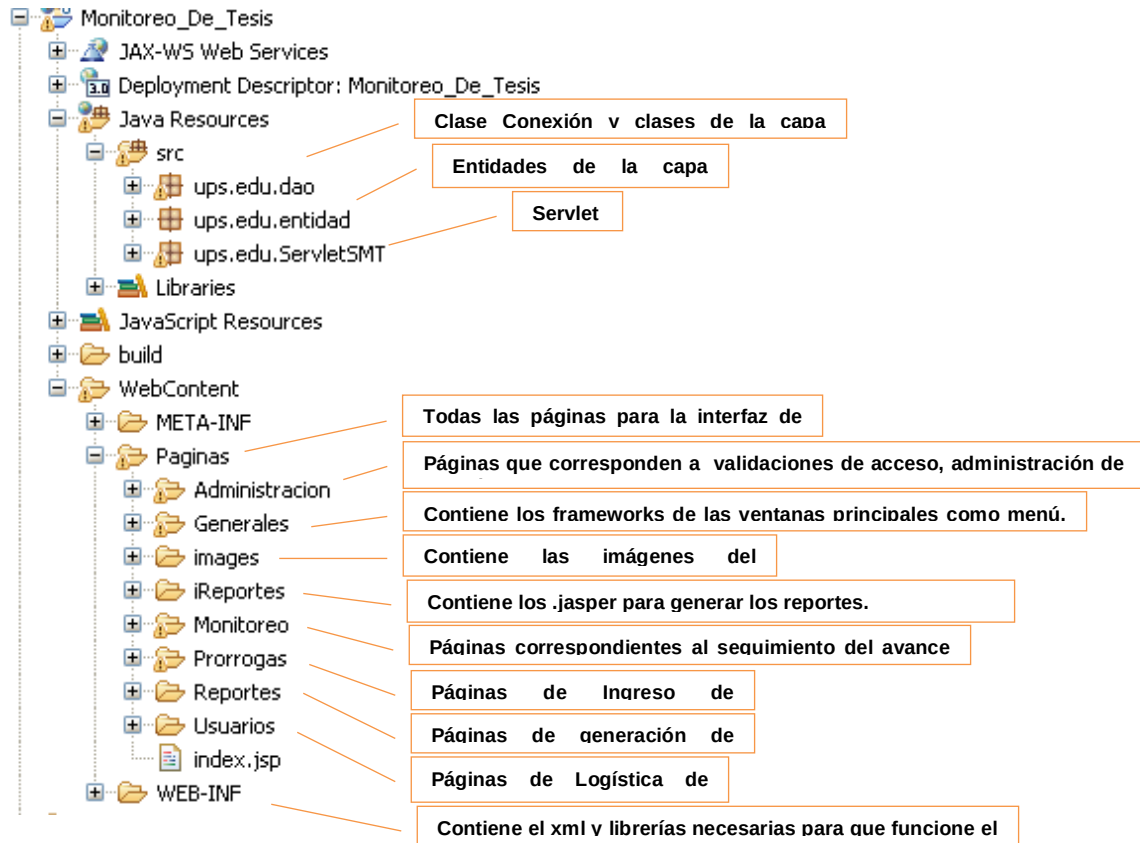
    JV.setVisible(true);

} catch (Exception e){
    this.releaseConnection();
}
}
```

3.3.4 UI (Interface User) Capa de presentación interfaz grafica

El SMT Web-Server (SISTEMA DE MONITOREO DE TESIS) cuenta con una interfaz web en un formato adecuado para la interacción de fácil comprensión brindando seguridad y comodidad para el usuario.

Descripción de archivos y carpetas importantes del sistema:



4. CONCLUSIONES

Con la investigación realizada a los estudiantes egresados de la Universidad Politécnica Salesiana se pudo demostrar que un gran porcentaje de estudiantes no presentan su tesis a tiempo, por motivo laborales y por lo que descuidan y olvidan su cronograma de actividades de tesis, por lo que fue necesario idear un sistema que permita realizar un seguimiento automático de este proceso y envíe recordatorios oportunos.

Para solventar el requerimiento de este sistema para la Universidad Politécnica Salesiana, se definió utilizar un motor con Base de Datos Oracle 10G que es el mismo motor de base de datos que soportan los sistemas de la institución.

El sistema realizó satisfactoriamente el envío de recordatorios a los estudiantes de la Universidad Politécnica Salesiana Sede Guayaquil por lo cual se evitarán olvidos de su cronograma y presentaciones de avances de tesis.

5. RECOMENDACIONES

Es recomendable que la información se maneje de una correcta manera para así descartar la posibilidad de pérdida de datos.

La asignación de usuarios debe realizarse de manera cuidadosa y únicamente por la persona asignada para administrar el sitio dado que este perfil se diseñó para una persona del departamento de Secretaría que es la encargada de lo que a tesis refiere, dado que este sistema accederá a información de la base de datos de la Universidad Politécnica Salesiana, la eliminación de los usuarios que ya no pertenecerán al grupo de tutores deberá realizarse de manera pertinente.

Se deberán otorgar todos los permisos necesarios especificados en las configuraciones para que el Sistema Automatizado de Monitoreo de Tesis pueda funcionar correctamente y se pueda acceder a la información necesaria, así como también los encargados del manejo de la información de la tesis deben ingresar la información propia de la automatización de manera oportuna para que se puedan aprovechar todas las facilidades y ventajas que ofrece el sistema.

Siempre se debe de tomar en cuenta que los equipos pueden sufrir accidentes en cuanto a su espacio destinado al almacenamiento, puede que sean fallas por interrupciones en el suministro eléctrico, sustracción de información por robo del dispositivo o una infección por un código malicioso, por lo tanto, realizar respaldos periódicos es una tarea que debe considerarse prioritaria y en ningún caso hay que subestimar las múltiples causas por las que podría ocurrir una situación de pérdida de información de lo referente a las tesis que es muy importante y esencial para el control de las mismas.

Bibliografía

CASTRO, E. (2007). *PROGRAMACION HTML,XHTMLYCSS*. ANAYA.

DEITEL, H., & DEITEL, P. (2004). *COMO PROGRAMAR EN JAVA* (QUINTA ed.). MEXICO: PEARSON.

LOPEZ QUIJADO, J. (2008). *DOMINE HTML Y DHTML* (SEGUNDA ed.). MEXICO: ALFAOMEGA.

Orellana, M. (2004). *HTML*. Guayaquil.

PEREZ, C. (2008). *ORACLE 10G ADMINISTRACION Y ANALISIS DEBASE DE DATOS* (SEGUNDA ed.). MEXICO: ALFAOMEGA.

CASAMAYOR, J., & MATAM, L. (2003). *BASE DE DATOS RELACIONALES*. MADRID: PEARSON

ANEXOS

Instalación de Oracle Database 10g Express Edition

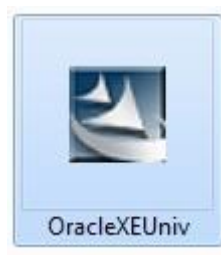
Antes de instalar Oracle 10g es necesario conocer los requerimientos de hardware y software, es posible desinstalar el software previamente, el chequeo del registro de Windows y el entorno del sistema, así como otras características de configuración más especializadas como pueden ser la reconfiguración de los servicios TCP/IP y la posible modificación de los tipos de ficheros HTML para navegadores como Netscape.

Requerimientos de hardware:

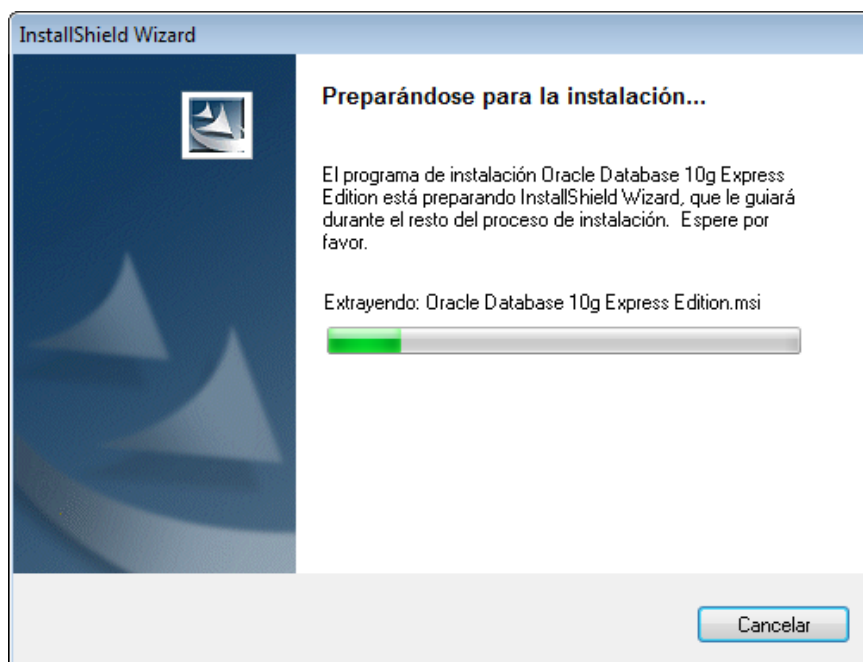
Una configuración mínima de hardware podría ser la siguiente:

- RAM:256 MB mínimo,512 recomendado
- Memoria virtual: Doble de la cantidad de RAM
- Espacio en disco duro: Se especificara a continuación
- Espacio para archivos temporales en disco 100MB
- Adaptadores de video :256 colores
- Procesador:200 MHz mínimo

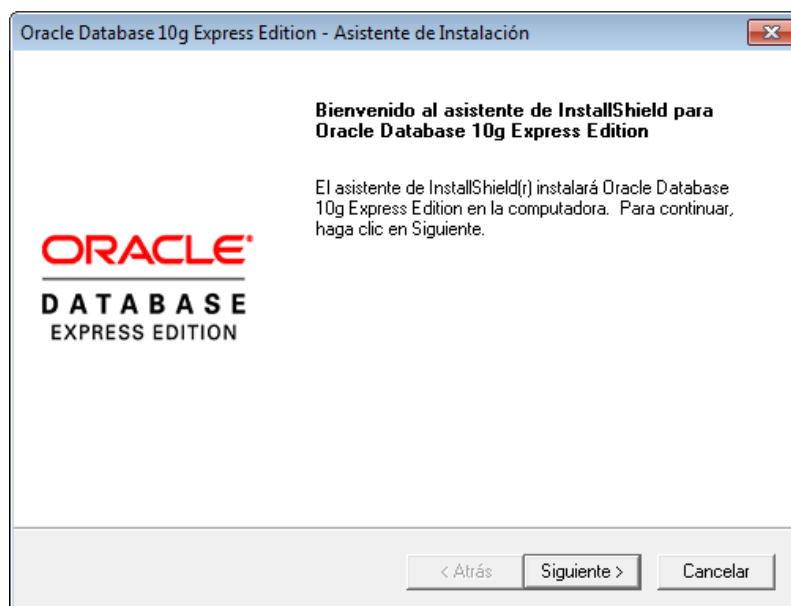
Seleccionamos el ejecutable, hacemos doble clic en el icono.



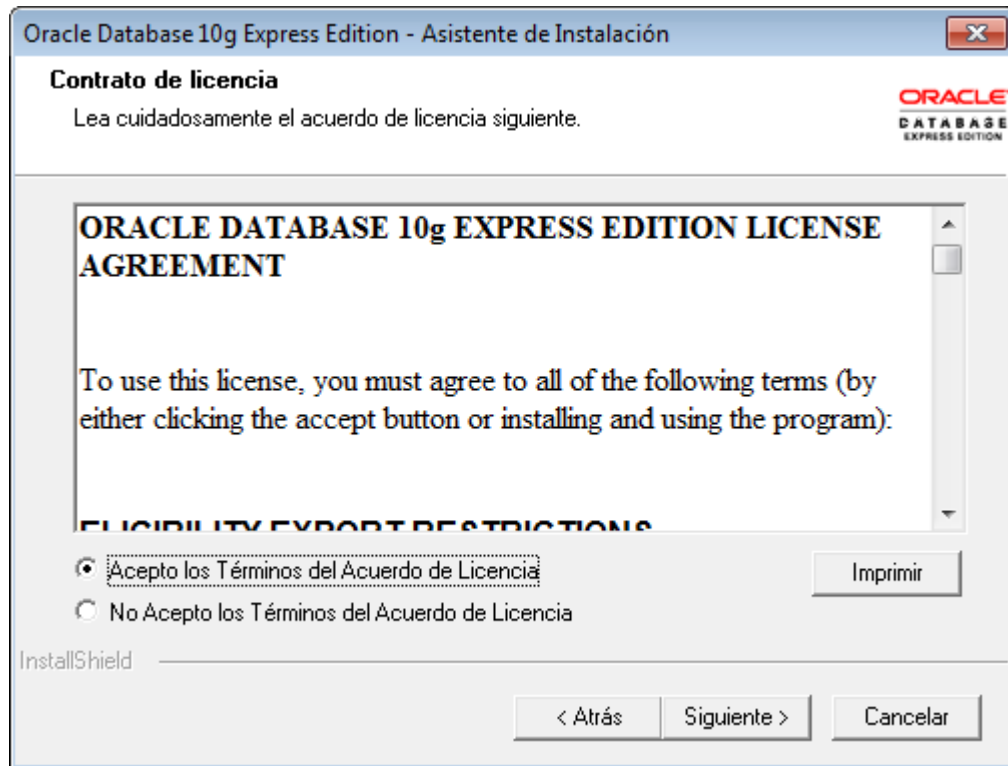
Una vez que hemos ejecutado el programa de instalación nos aparecerá esta primera ventana donde podemos apreciar que el programa de instalación de Oracle Database 10g Express Edition XE está preparando el wizard para la instalación.



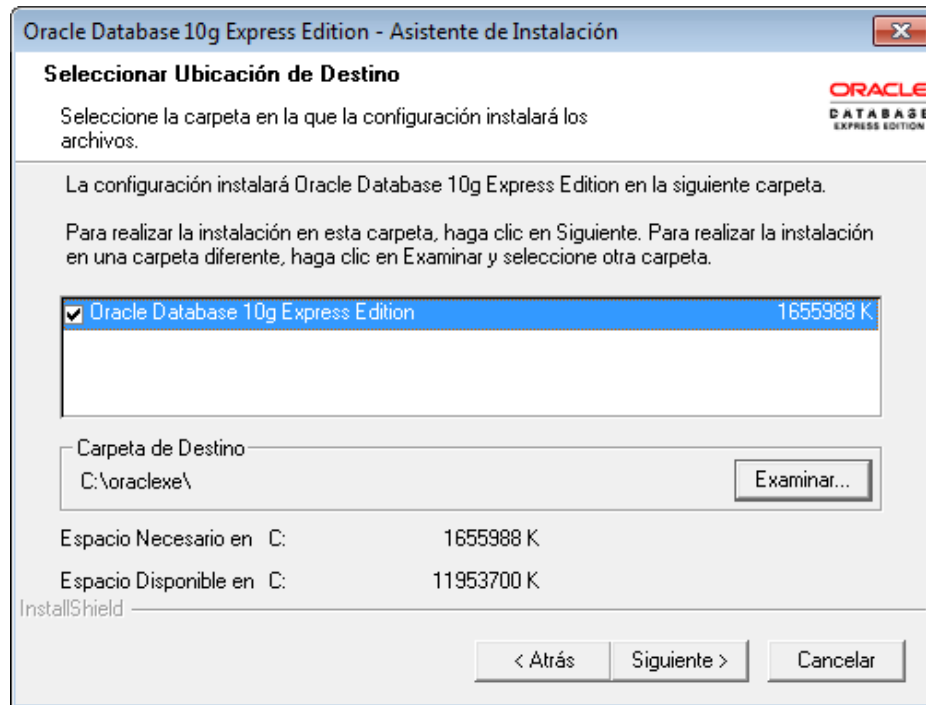
Aquí tenemos la primera ventana del wizard de la instalación de Oracle Database 10g Express Edition XE, pulsamos **Siguiente >** para continuar.



Contrato de licencia seleccionamos *–Acepto los términos del Acuerdo de Licencia,*
pulsamos para continuar.

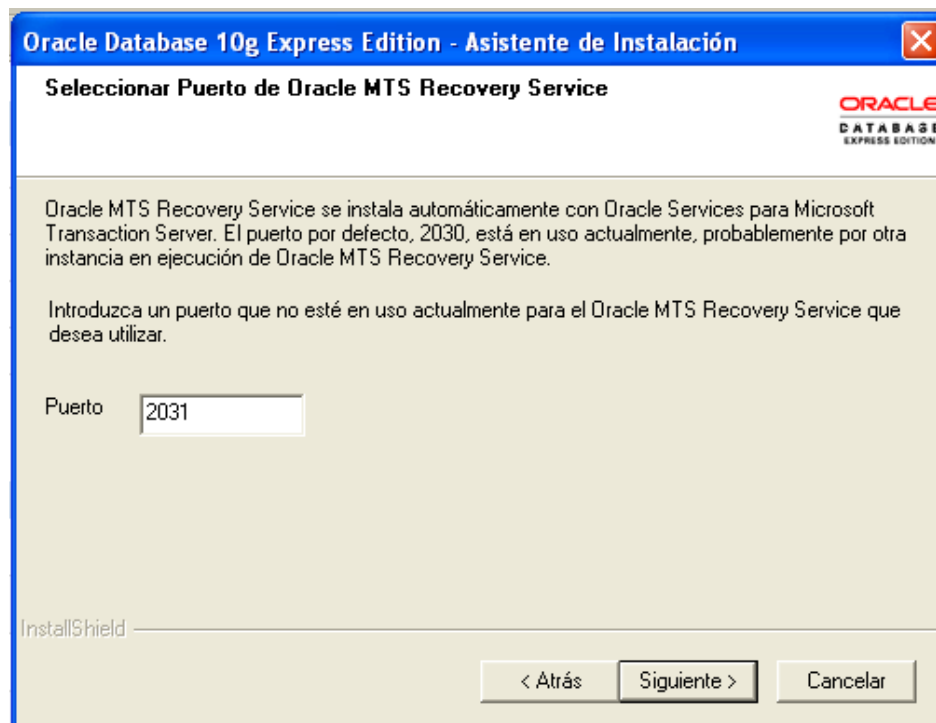


En esta ventana el Wizard nos presenta la carpeta destino para la instalación por defecto así como el espacio necesario y el disponible, si desea cambiarla pulsar en el botón y seleccionar la carpeta donde desea realizar la instalación. Una vez cambiada o si la ubicación por defecto es válida pulsamos para continuar.



En esta ventana el instalador nos propone para Oracle MTS el puerto 2030 por defecto, en el caso de que este ocupado propone un puerto que este libre, pulsamos el botón

Siguiente >



Esta ventana es muy importante tenemos que introducir la contraseña que se usará para los usuarios SYS y SYSTEM, guardarla bien ya que si esta contraseña la olvida no podrá realizar tareas de administración.

Una vez hayamos introducido la contraseña y la confirmación pulsamos para continuar.

Oracle Database 10g Express Edition - Asistente de Instalación

Especificar Contraseñas de Base de Datos

Introduzca y confirme las contraseñas para la base de datos. Esta contraseña se utilizará tanto para la cuenta de base de datos SYS como para la cuenta SYSTEM.

Introducir Contraseña

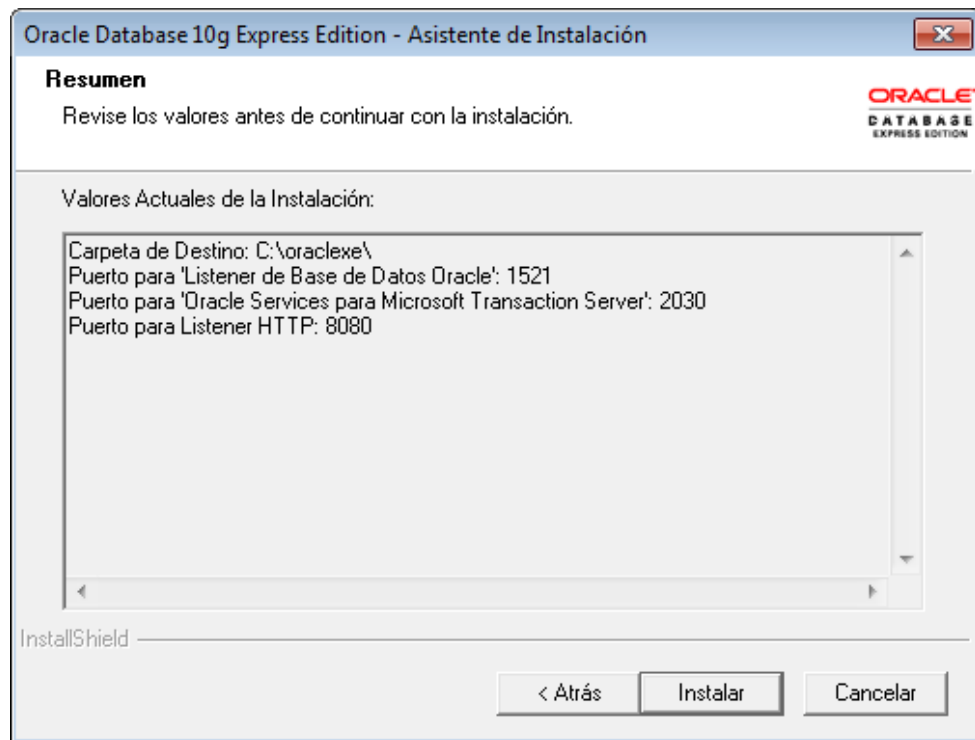
Confirmar Contraseña

Nota: Debe utilizar el usuario SYSTEM junto con la contraseña introducida aquí para conectarse a la página inicial de la base de datos después de terminar la instalación.

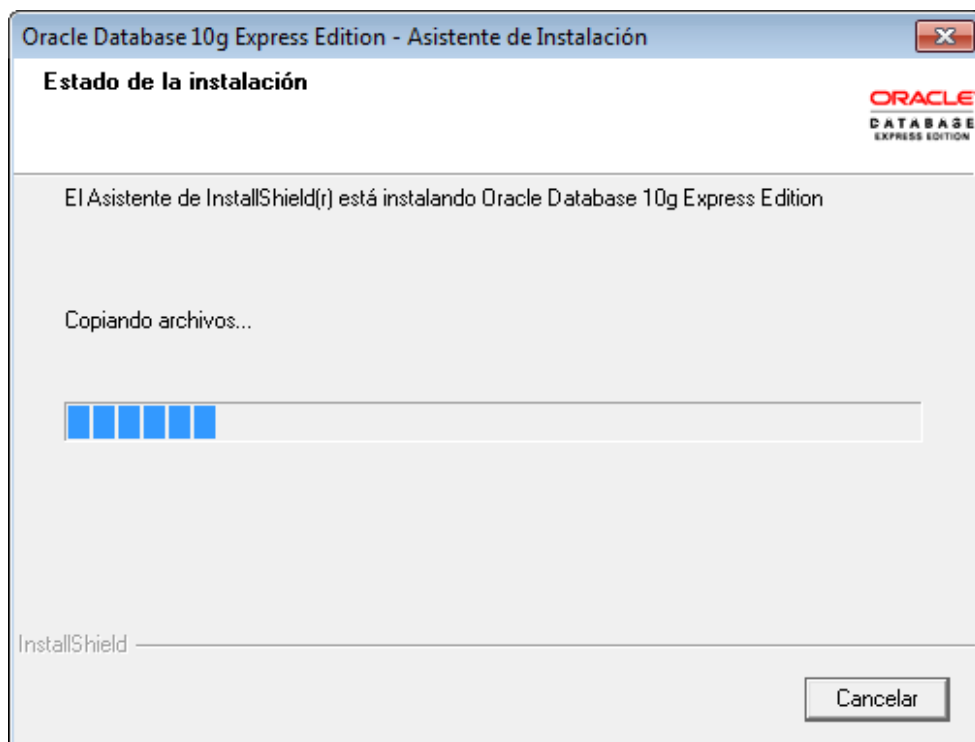
InstallShield

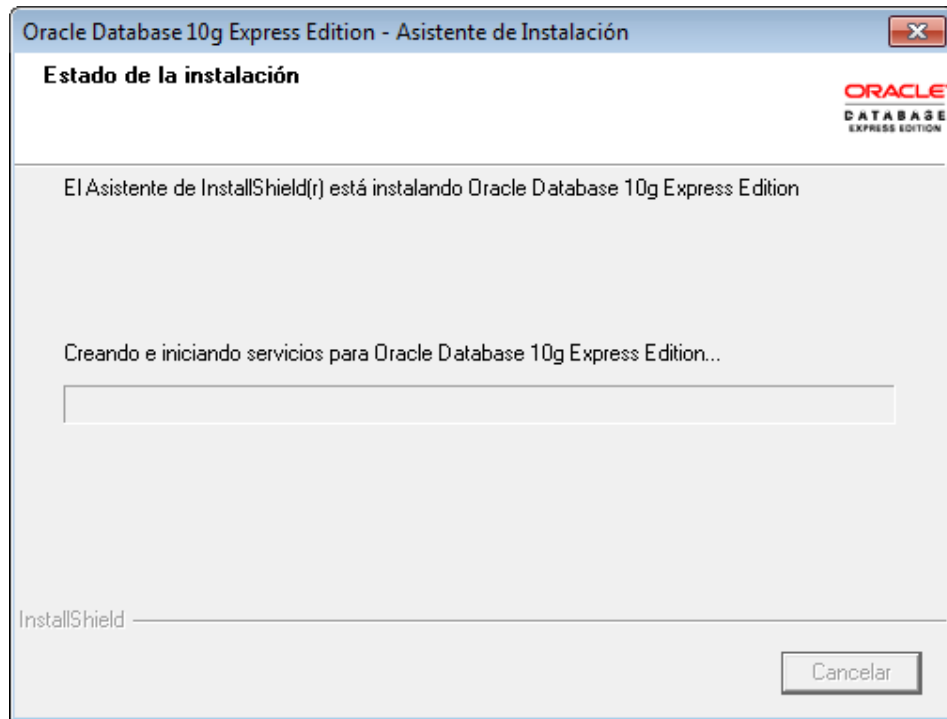
El Wizard ya está preparado para iniciar la instalación nos muestra un resumen de los valores que va a utilizar para el Listener. Todos estos puertos son los valores por defecto que Oracle usa en cualquiera de sus productos

Pulsamos para iniciar la instalación.

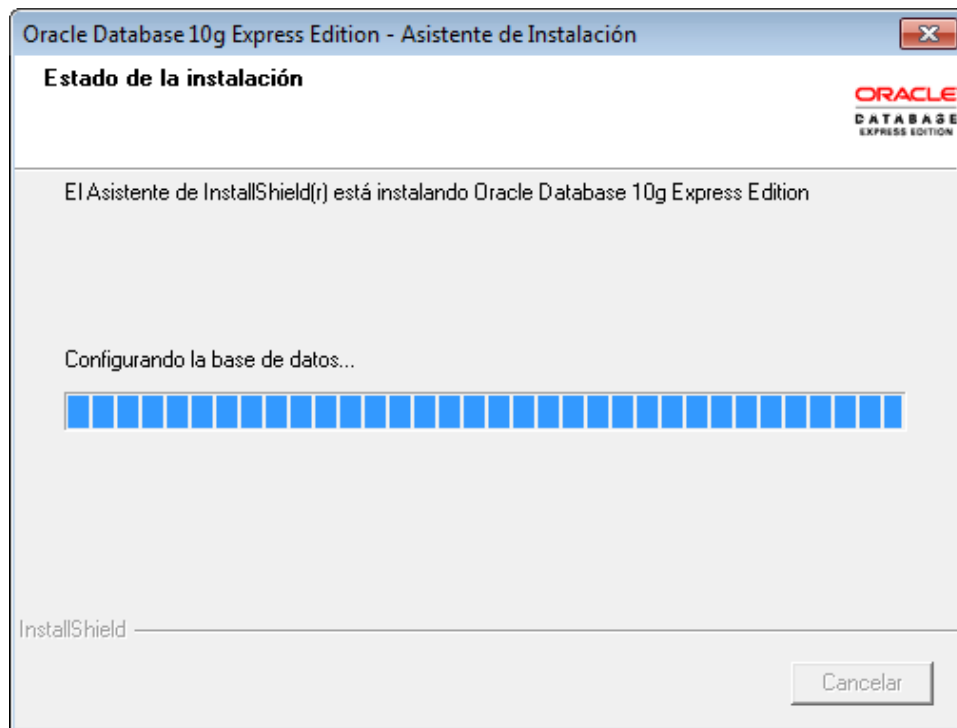


Empezaremos a ver una serie de ventanas que nos irá informando del progreso de la instalación.



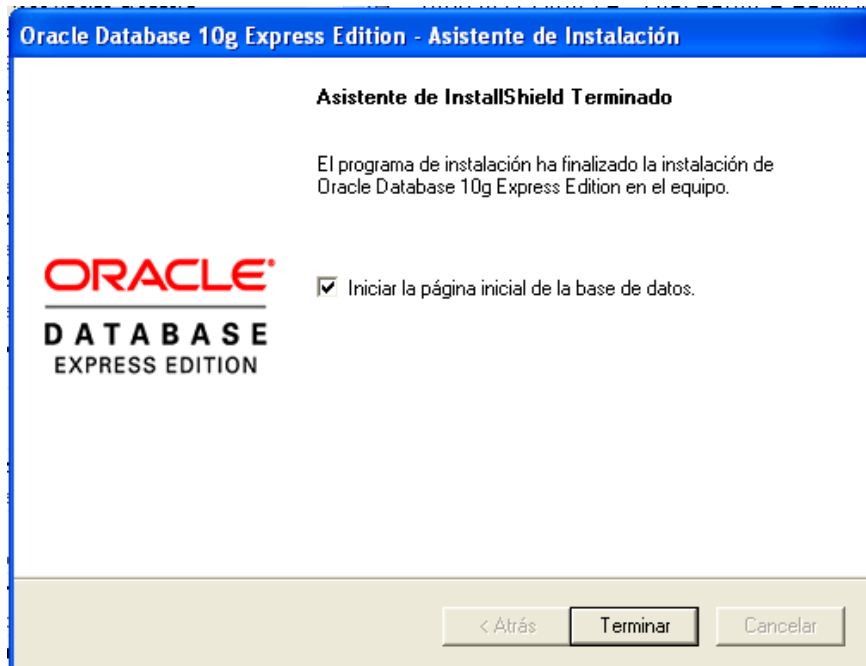


En esta última ventana el Wizard realiza la creación y configuración de la base de datos.



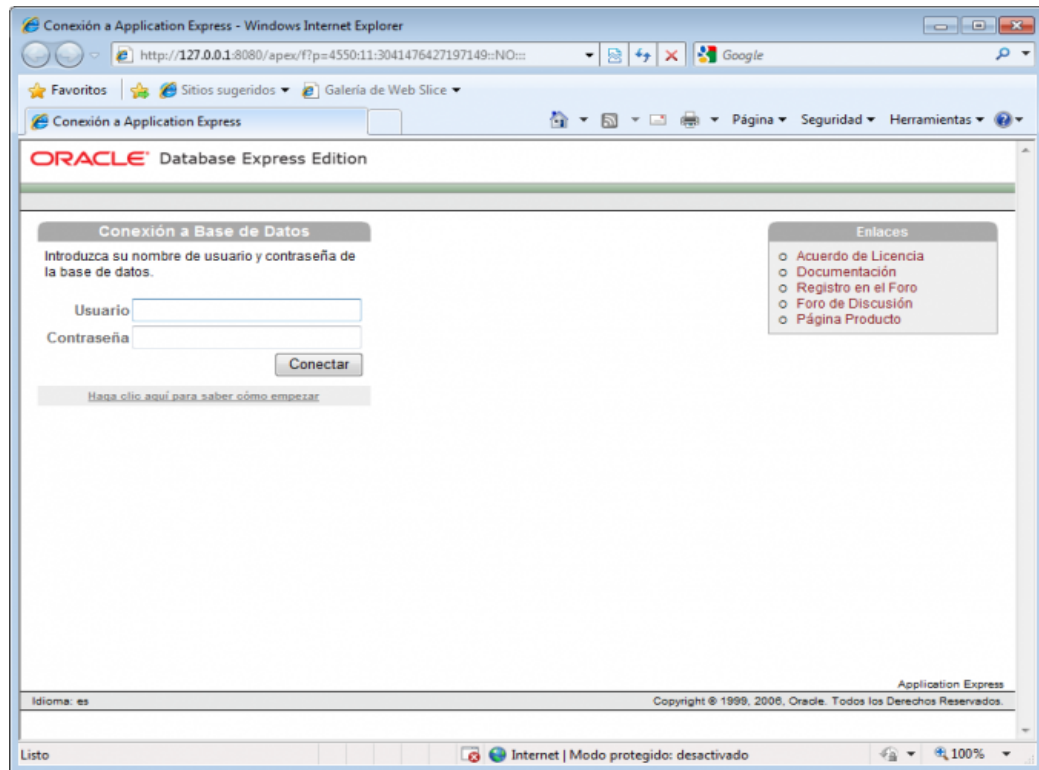
Finalmente la instalación termina, dejamos seleccionado – *Iniciar la página inicial ...* -

y pulsamos el botón 

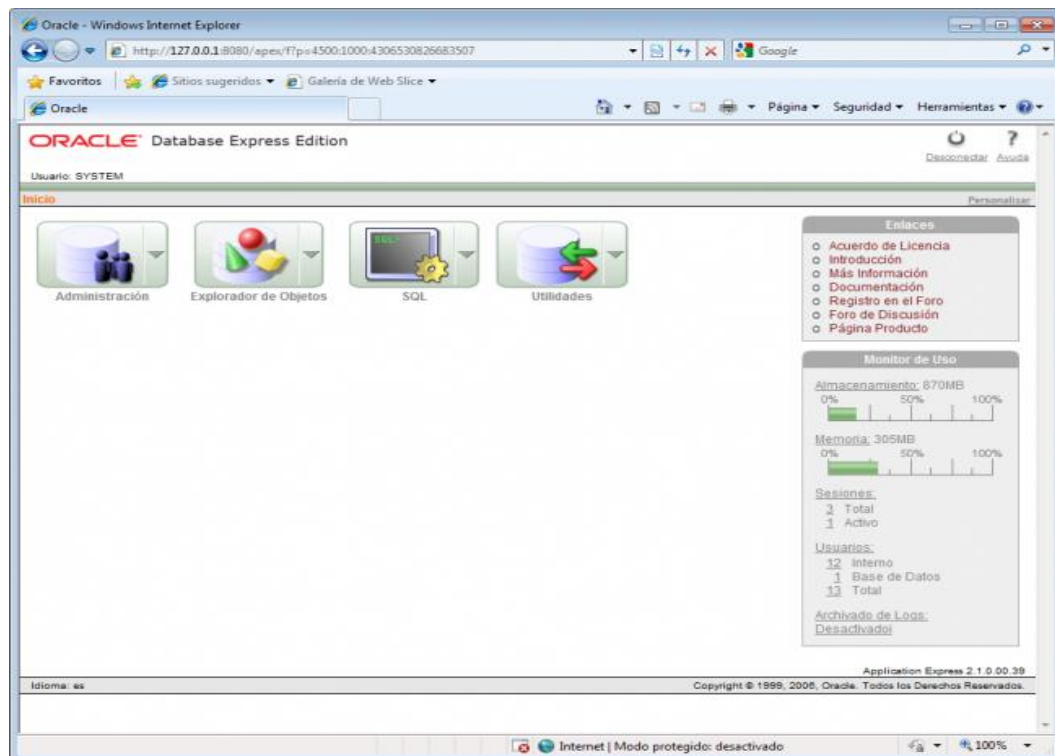


Tenemos el interface web para conectarnos a la base de datos.

Nos logueamos con el usuario SYSTEM y la contraseña que hayamos introducido en el proceso de instalación.

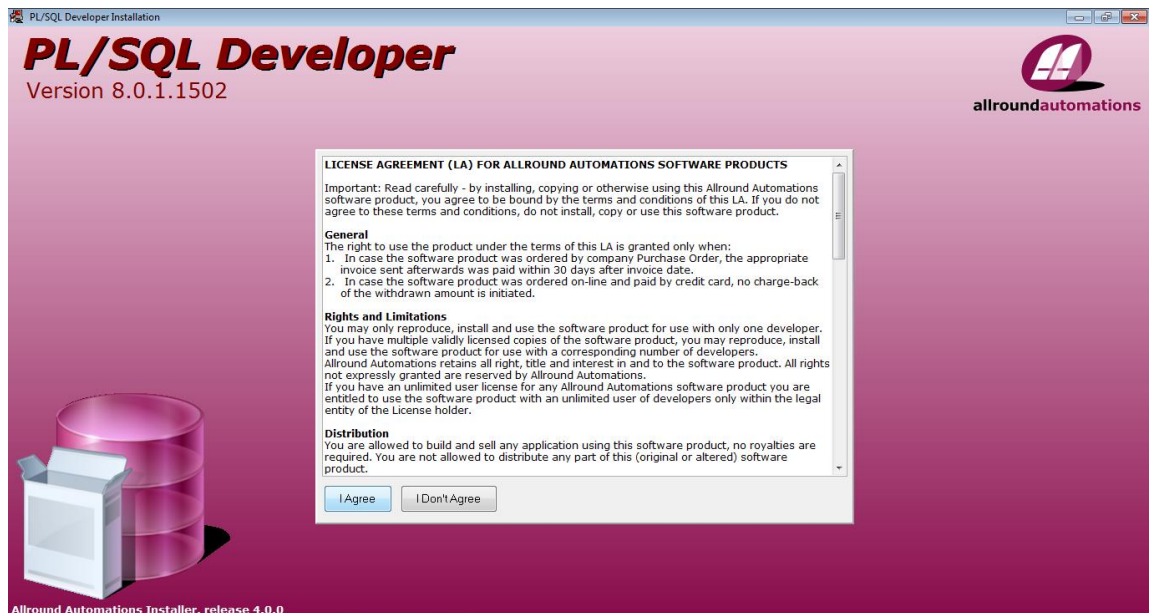
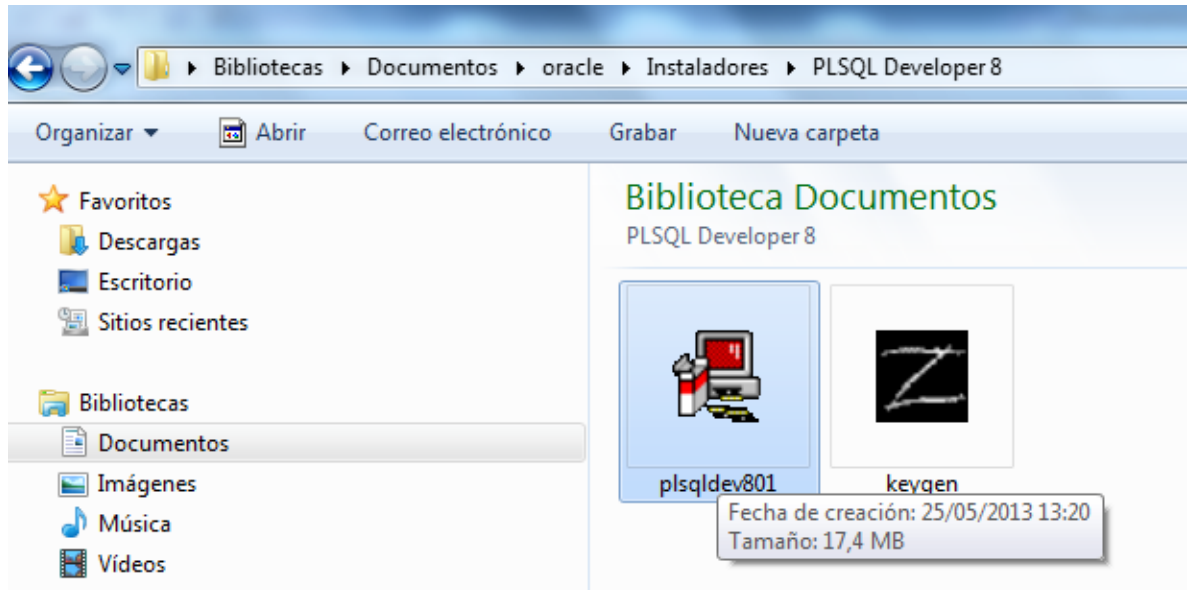


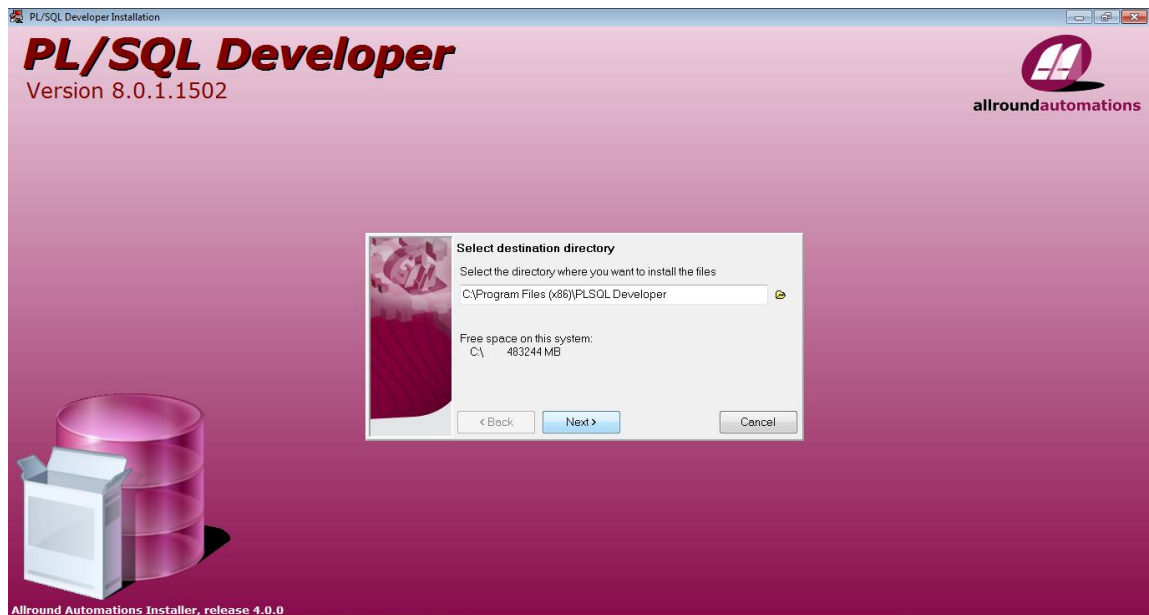
Nos aparece el navegador con la siguiente presentación, desde esta pantalla podemos administrar la base de datos, gestionar objetos, etc.



Instalación de PLSQL DEVELOPER 8

Le damos doble clic sobre el icono para comenzar la instalación.

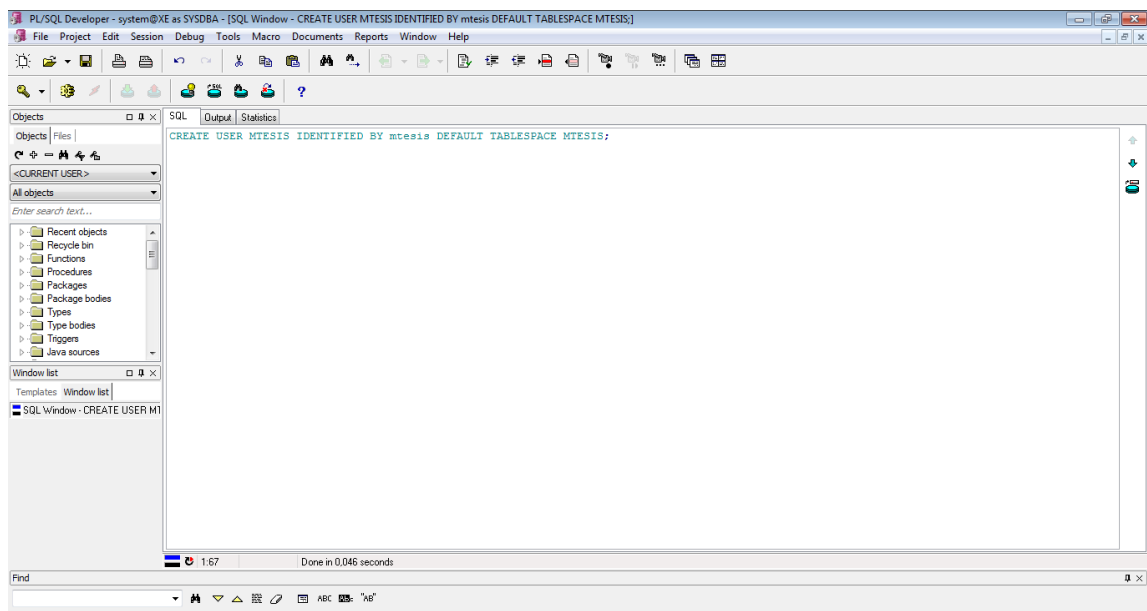
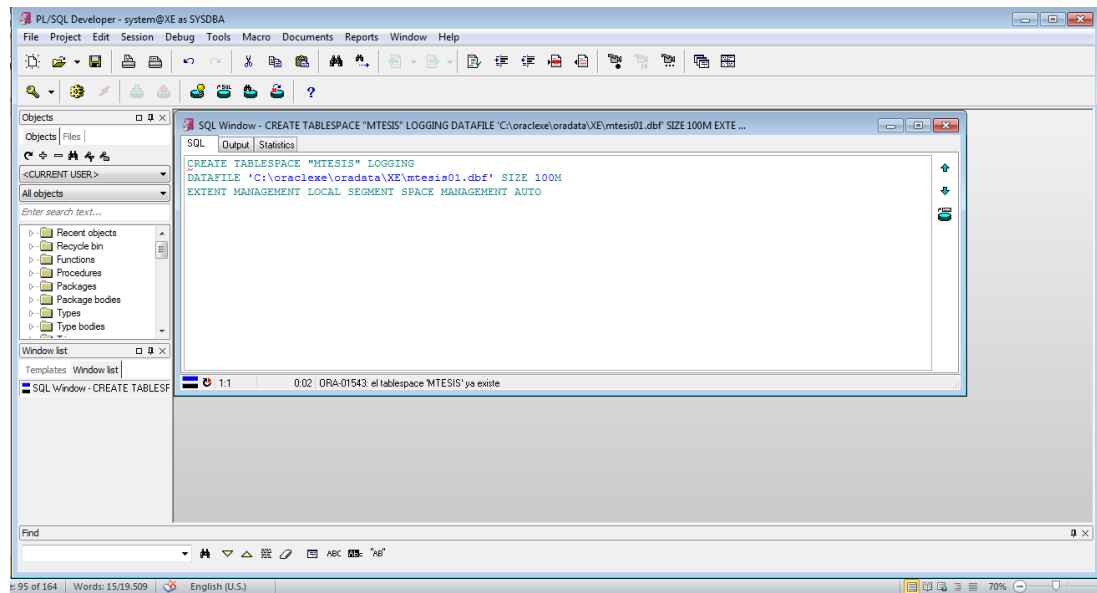


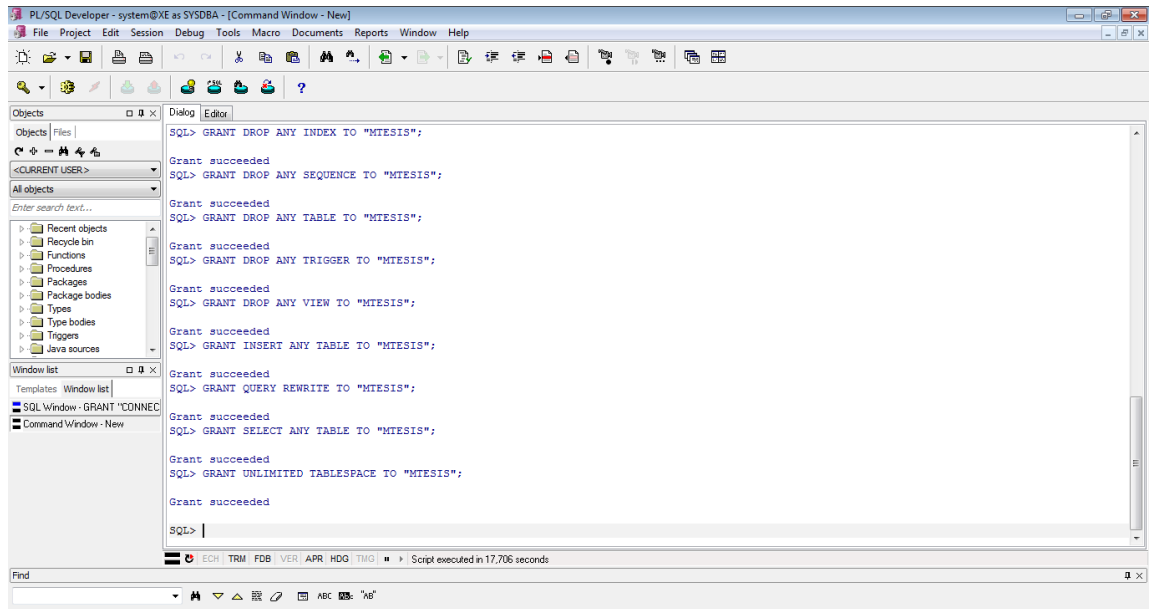


Una vez instalados ingresamos con el usuario System y la contraseña ya configurado en la instalación de Oracle.



Una vez que hemos ingresado creamos el tablespace de mtesis y el usuario para agregar las tablas de nuestro sistema.

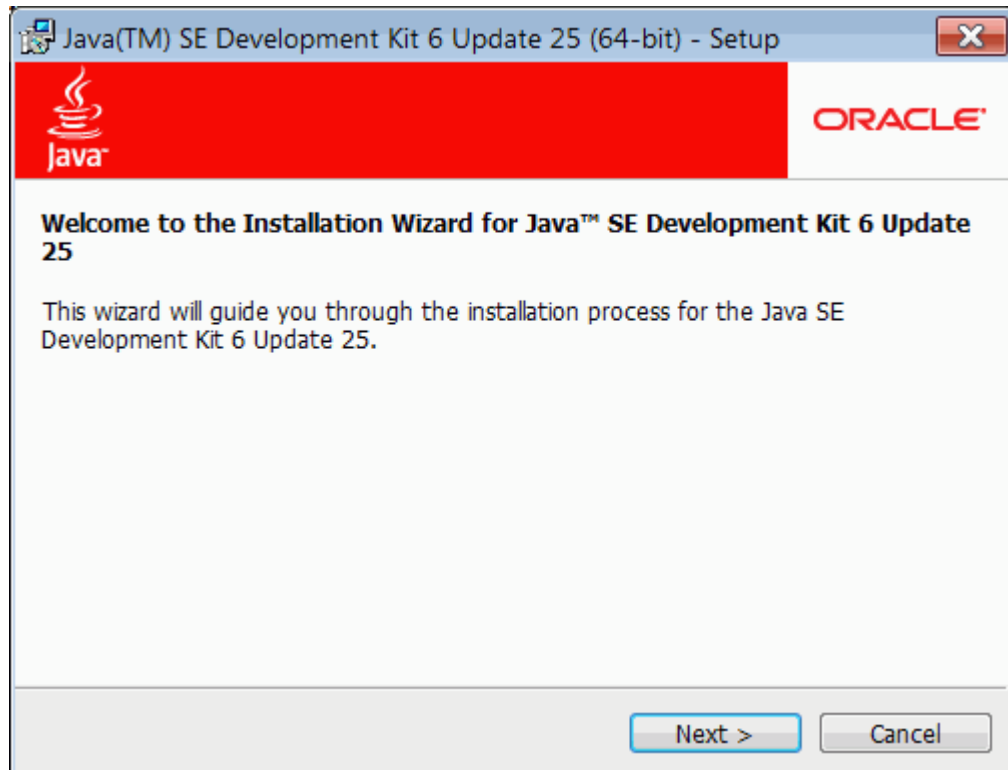




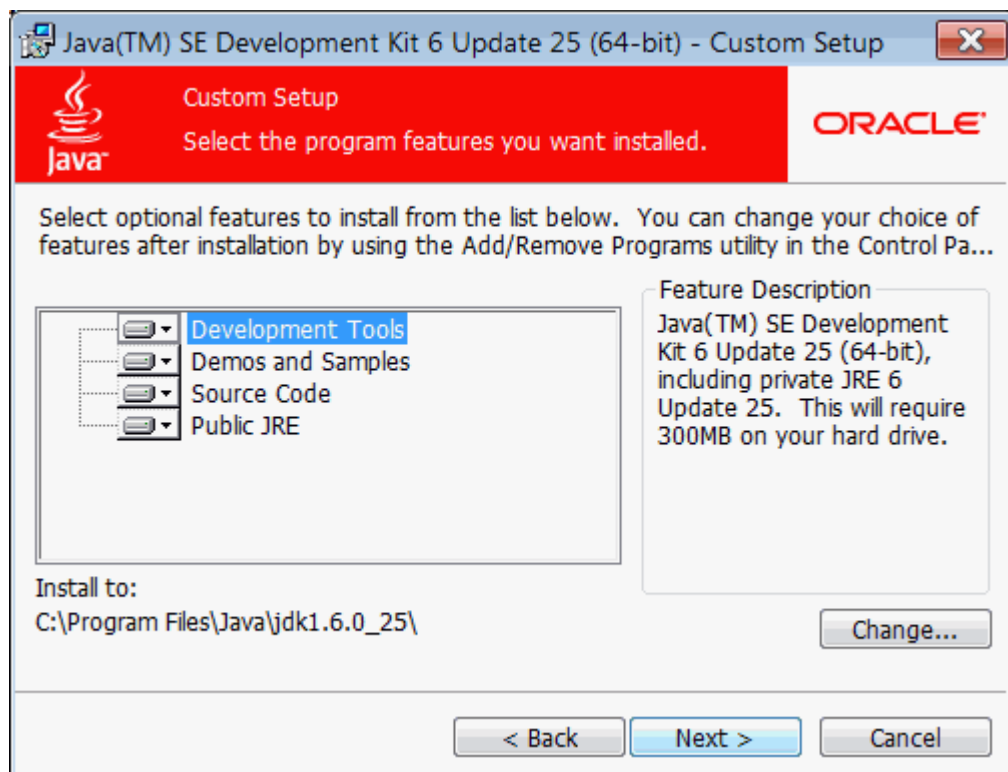
Instalación de JDK

Seleccionamos el ejecutable, hacemos doble clic en el icono, para comenzar la instalación.

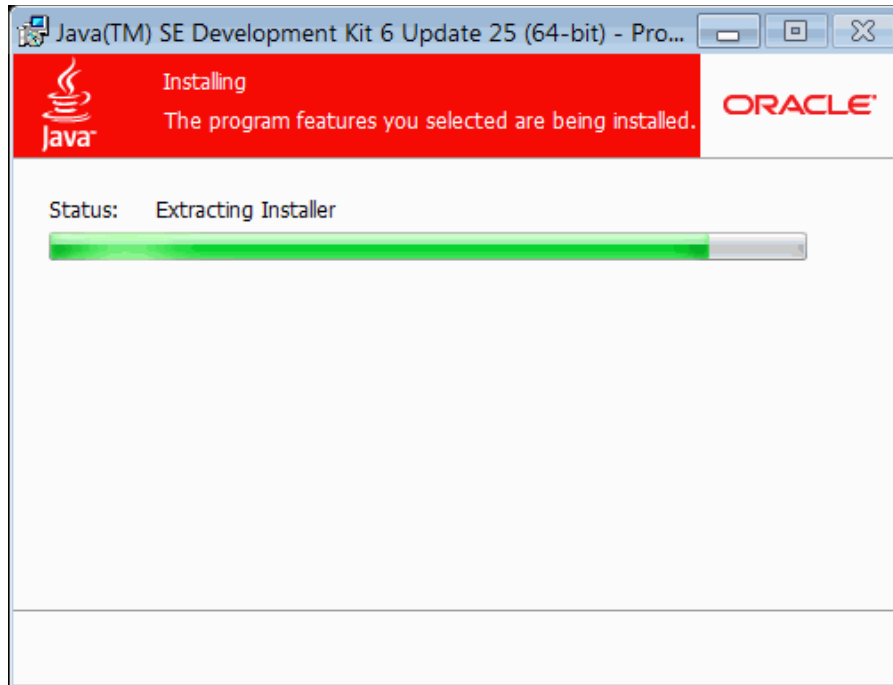




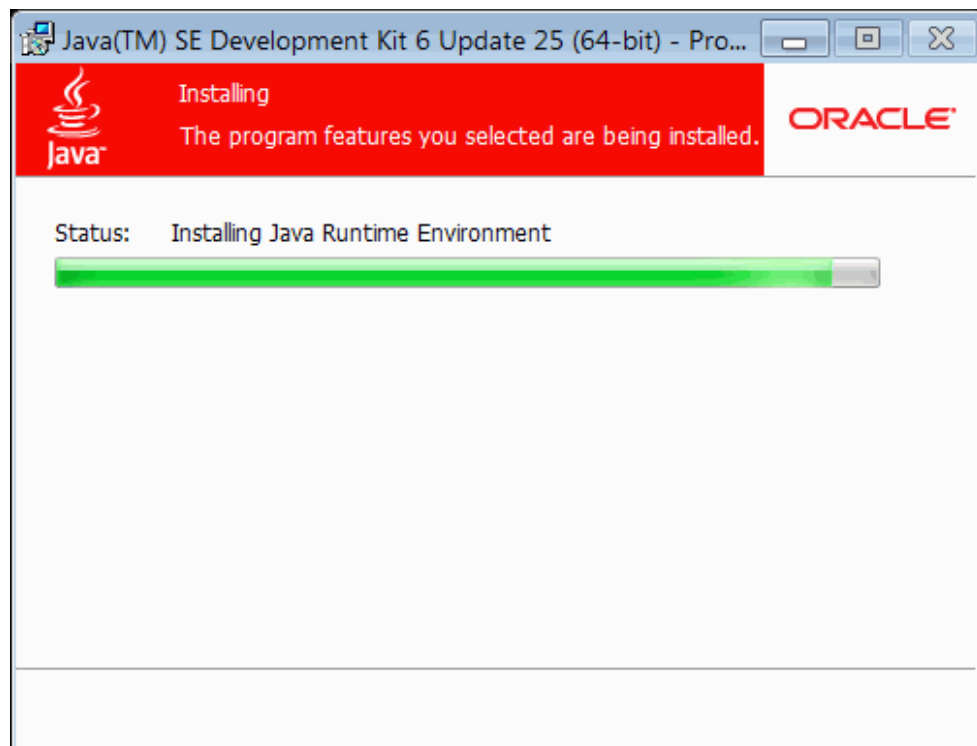
Si tenemos suficiente espacio (posiblemente unos 400 Mb en total), podemos hacer una instalación típica:



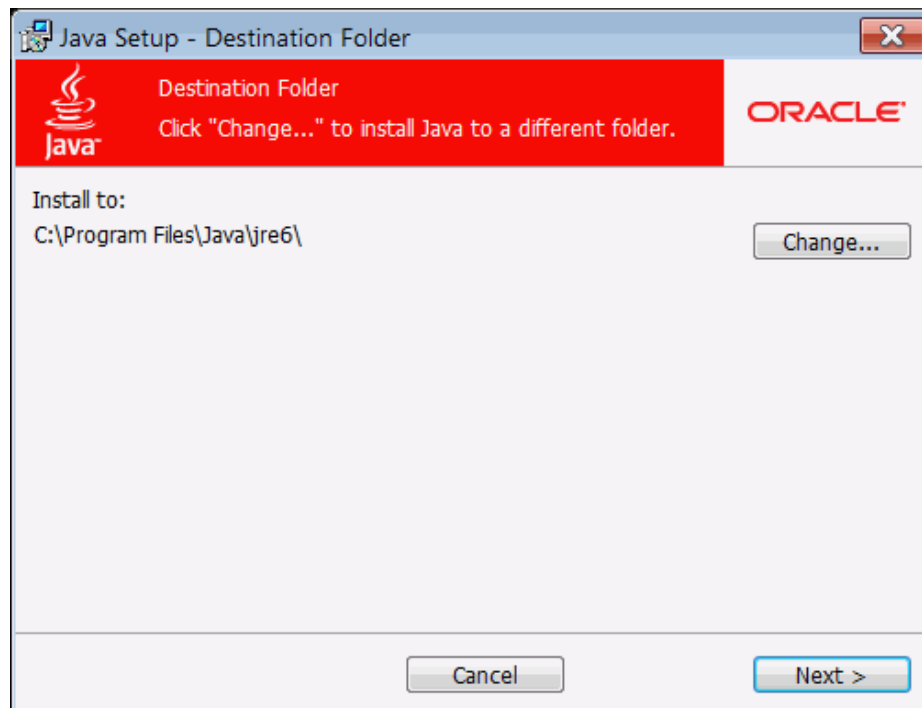
Ahora tendremos que tener paciencia durante un rato, mientras se descomprime e instala todo:



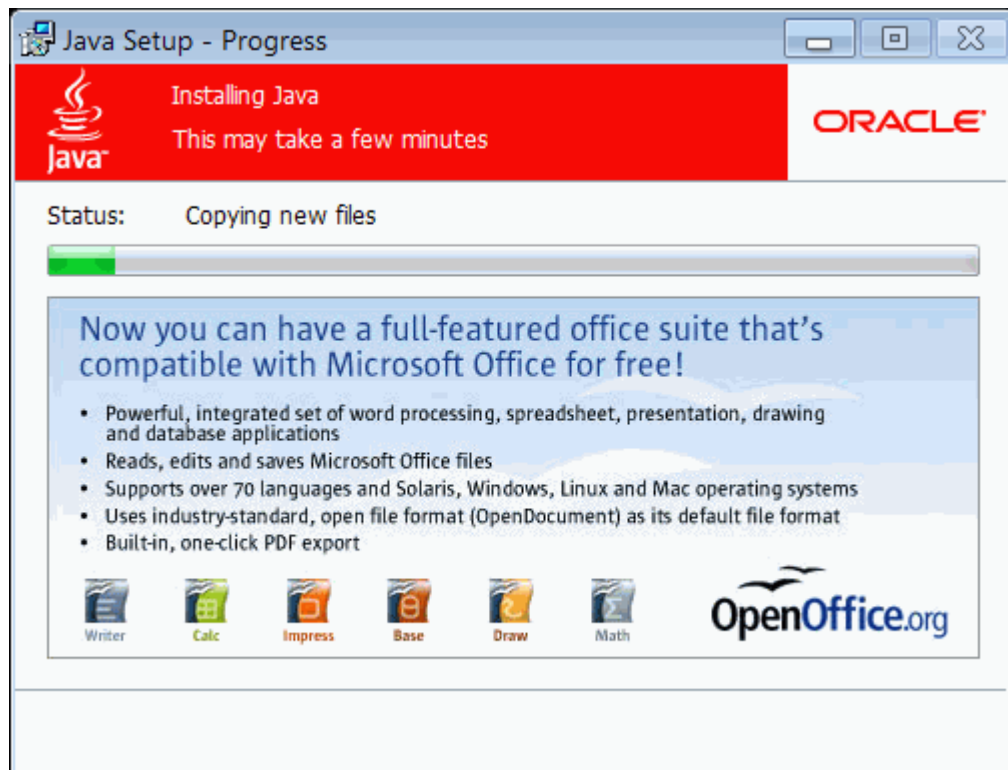
En cierto punto se nos preguntará si queremos instalar la máquina virtual Java (Java Runtime Environment, JRE). Deberá seleccionar que sí:



Igual que para el JDK, podríamos cambiar la carpeta de instalación:



Tendremos que esperar otro momento

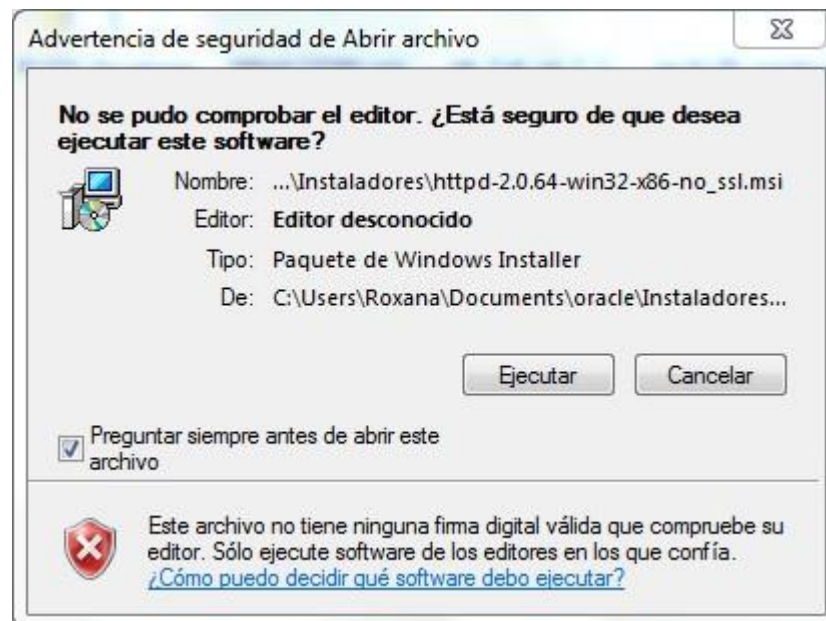


Si todo ha ido bien, deberíamos obtener un mensaje de confirmación:



Instalación Apache

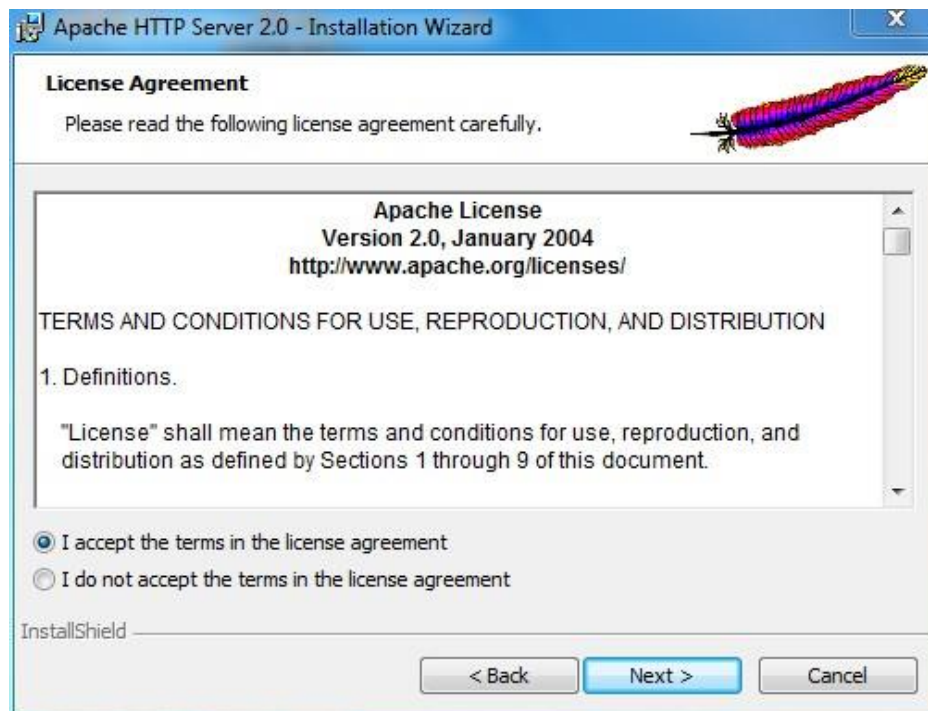
Damos doble click en el instalador y click en Ejecutar



La primera ventana que nos aparece es la de "Welcome to the Installation Wizard for Apache HTTP Server 2.2.17", click **Next**.



Una vez en los Términos y Condiciones de uso, leemos la licencia, y damos un click en I accept the terms in the license agreement para poder continuar y luego click en Next.



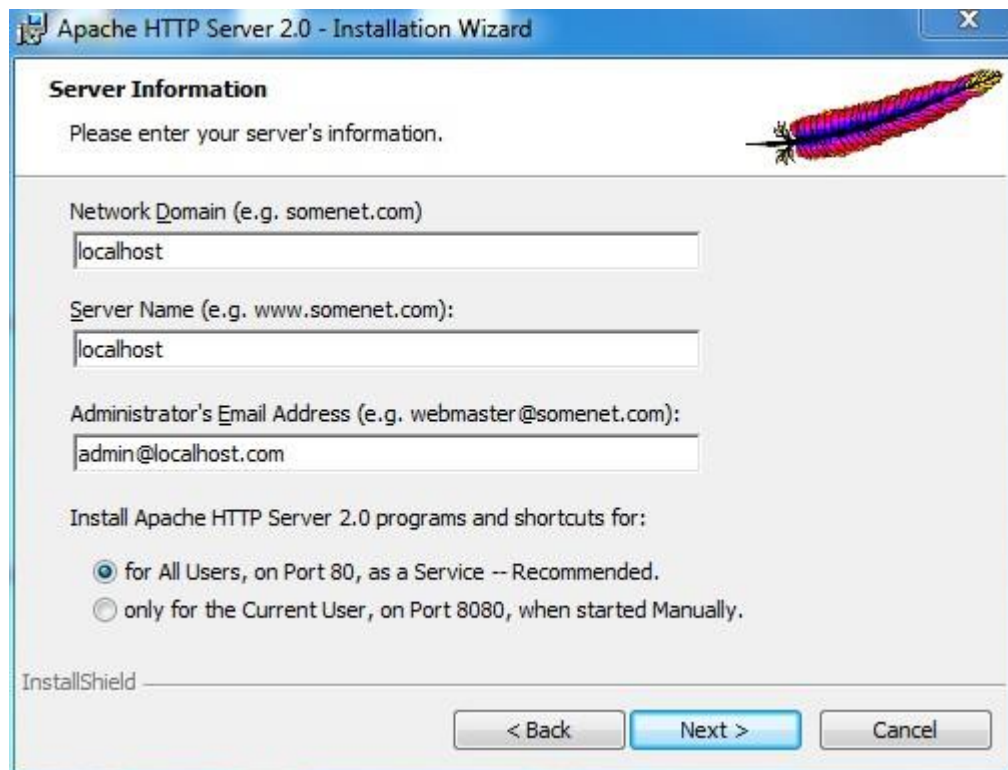
En este paso recomiendo leer bien la información proporcionada por el instalador de Apache, ya que ésta es de importancia para el buen uso del servidor HTTP de Apache, la cual incluye la ubicación de la documentación como en <http://httpd.apache.org/docs/2.2>. Luego de leer, click en Next.



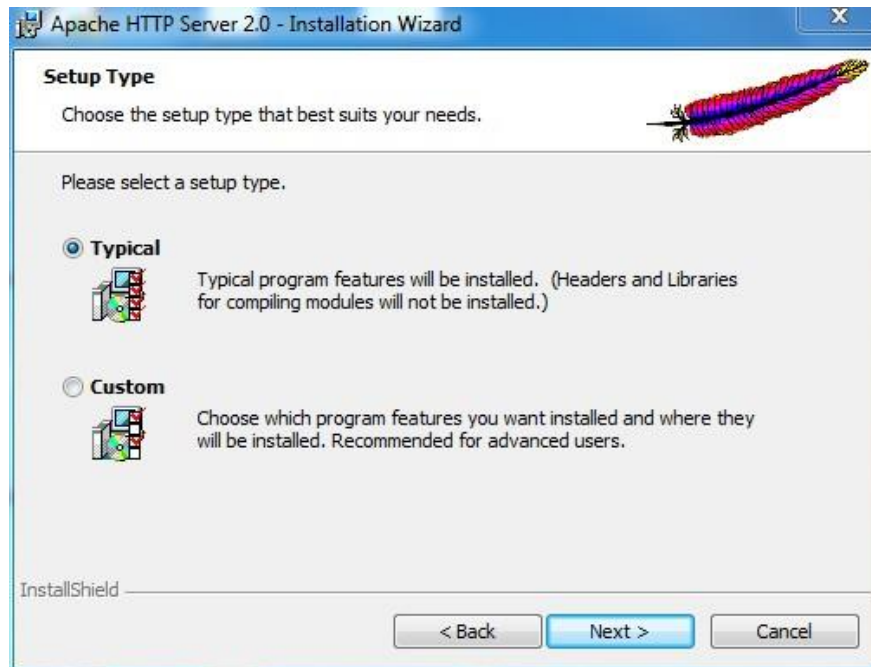
Ahora en este paso de la instalación ingresamos la información del servidor Apache. Como usaremos el servidor Apache en un entorno local para nuestras pruebas escribimos lo siguiente:

- Network Domain: localhost
- Server Name: localhost
- Administrator's Email: cualquier@email.com

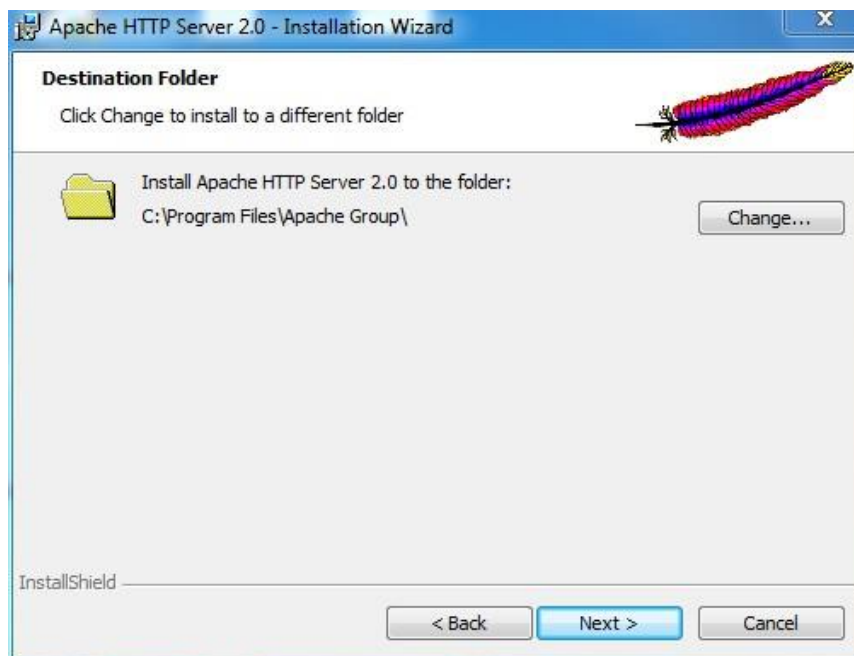
Estos datos los podemos cambiar . Luego recomiendo escoger la opción *for All Users*, on port 80, as a *Service -- Recommended*. Esta opción dejará nuestro servidor Apache corriendo como un servicio más de windows cada vez que iniciemos nuestro pc. Click en Next.



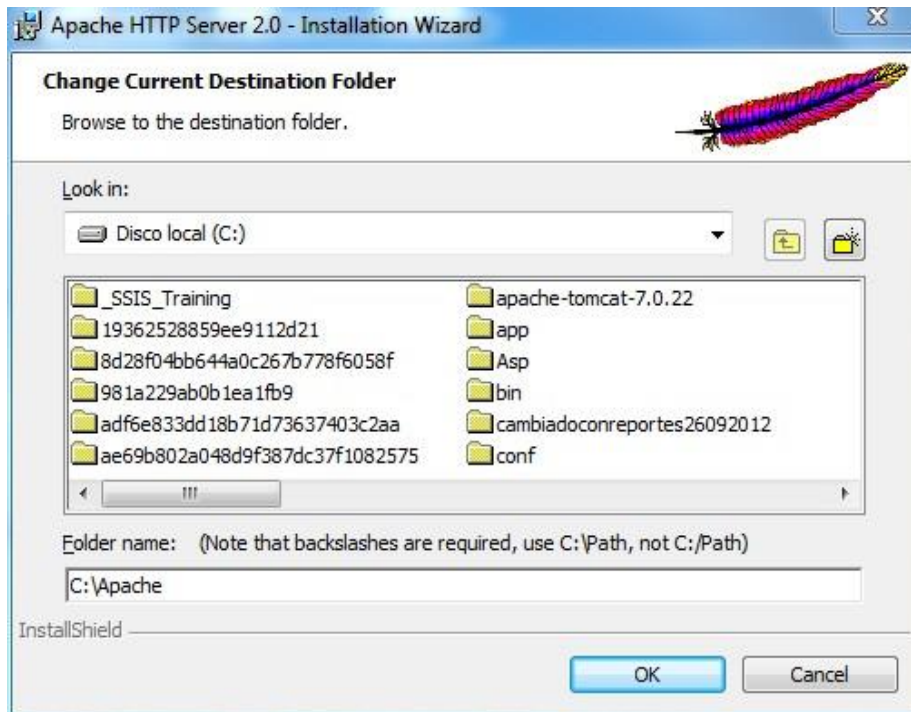
Seleccionamos la opción Typical Click en Next para continuar.



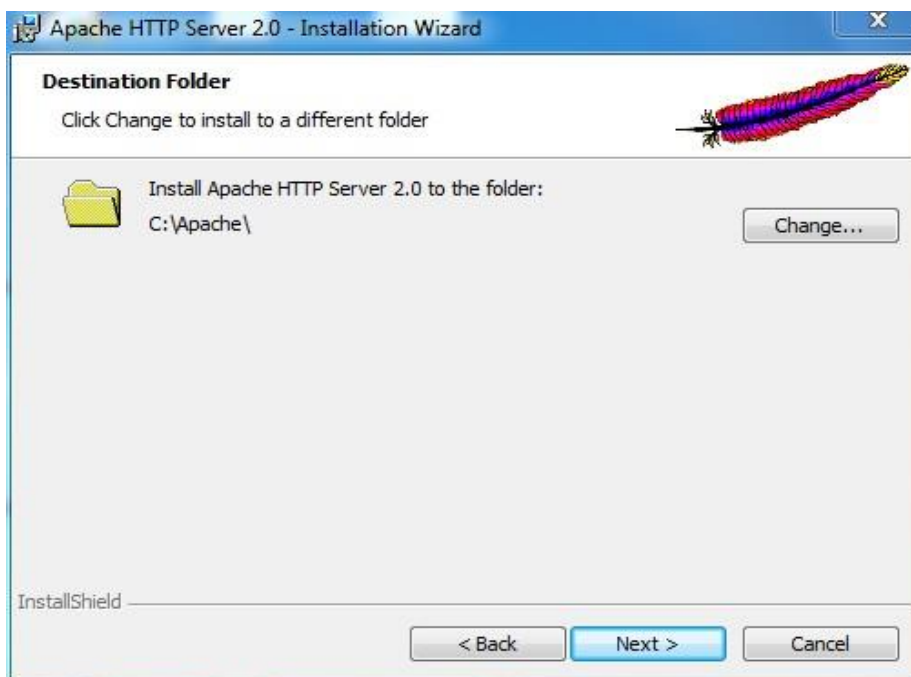
En este paso el instalador de Apache nos sugiere la ubicación en donde se instalará el servidor HTTP de Apache. Recomendando cambiar la ubicación a C:\apache ya que esto nos permitirá trabajar con Apache fácilmente y tendremos los archivos fuera de nuestra carpeta de sistema. Click en Change...



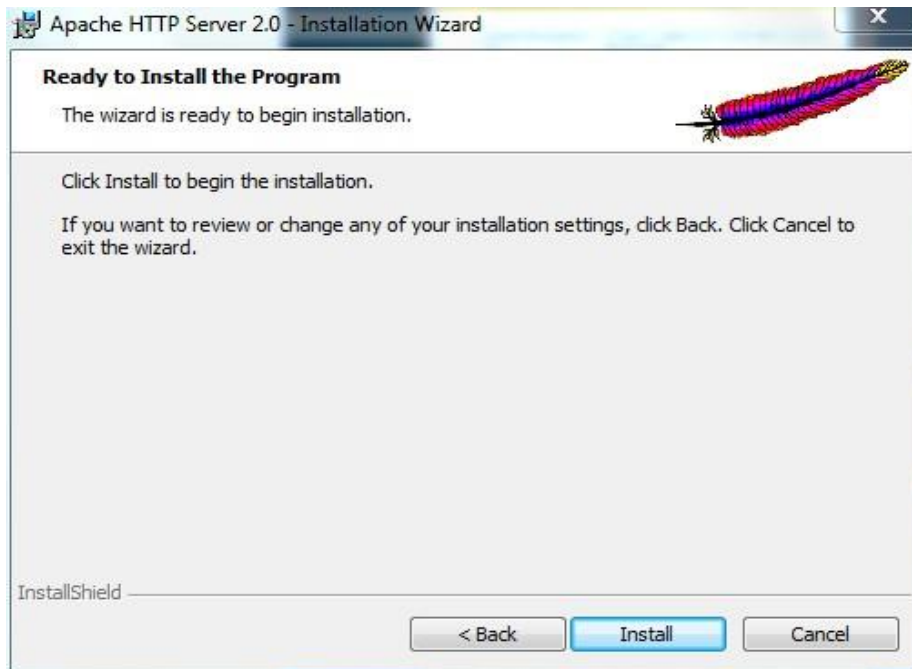
En Folder name escribimos C:\apache y damos click en OK.



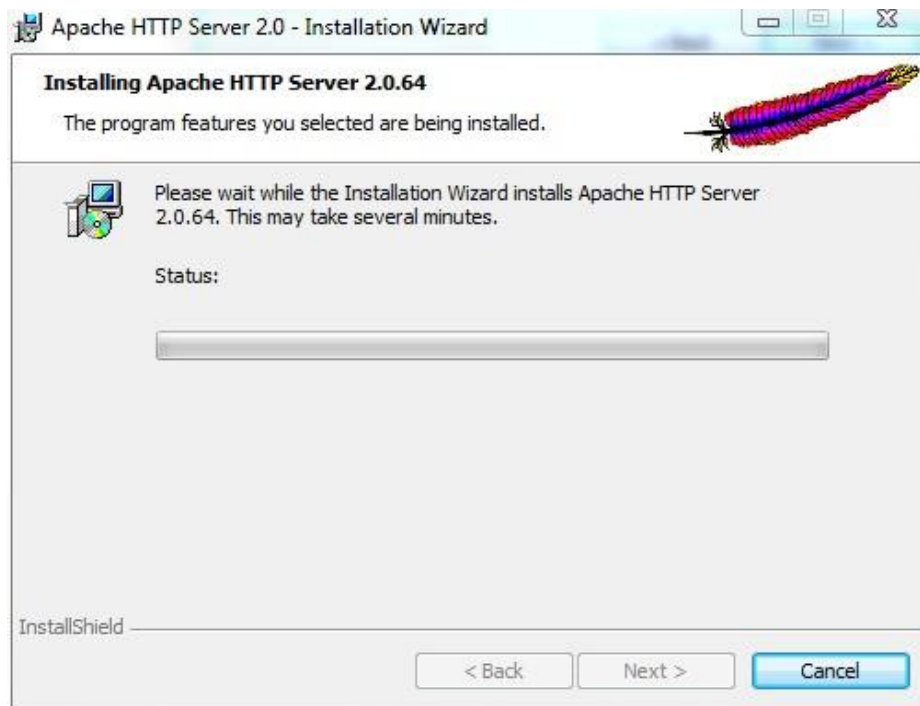
Una vez tenemos la nueva ubicación c:\apache en donde se instalará el servidor http de Apache continuamos dando un click Next.



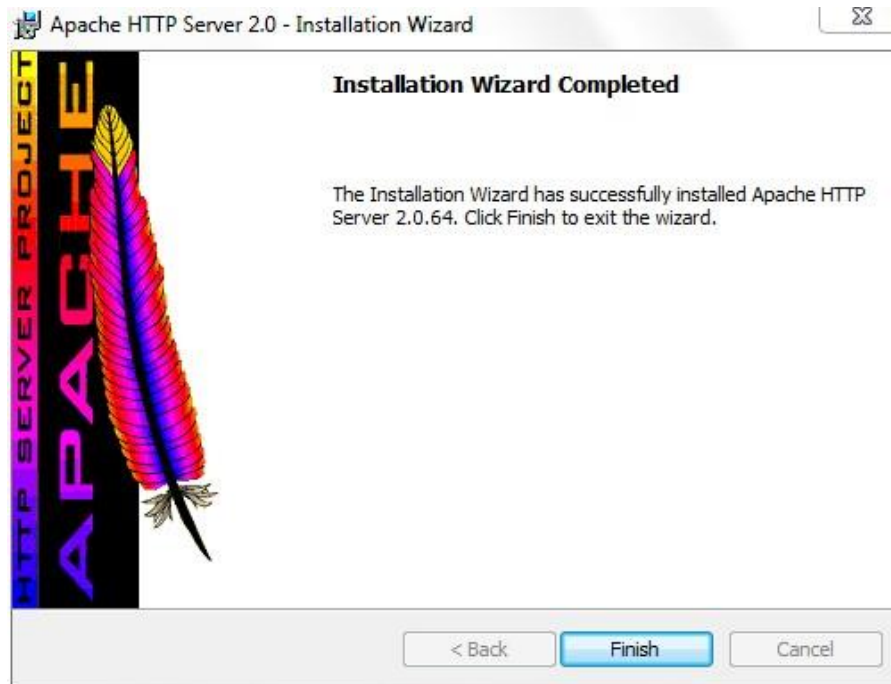
En este paso ya tenemos toda la configuración lista. Un click en Install y todo sera instalado.



Esperamos un par de minutos a que termine la instalación del servidor http de Apache.



En este paso una vez terminada la instalación del servidor http de Apache sin ningún error cerramos la ventana dando un click en **Finish**.

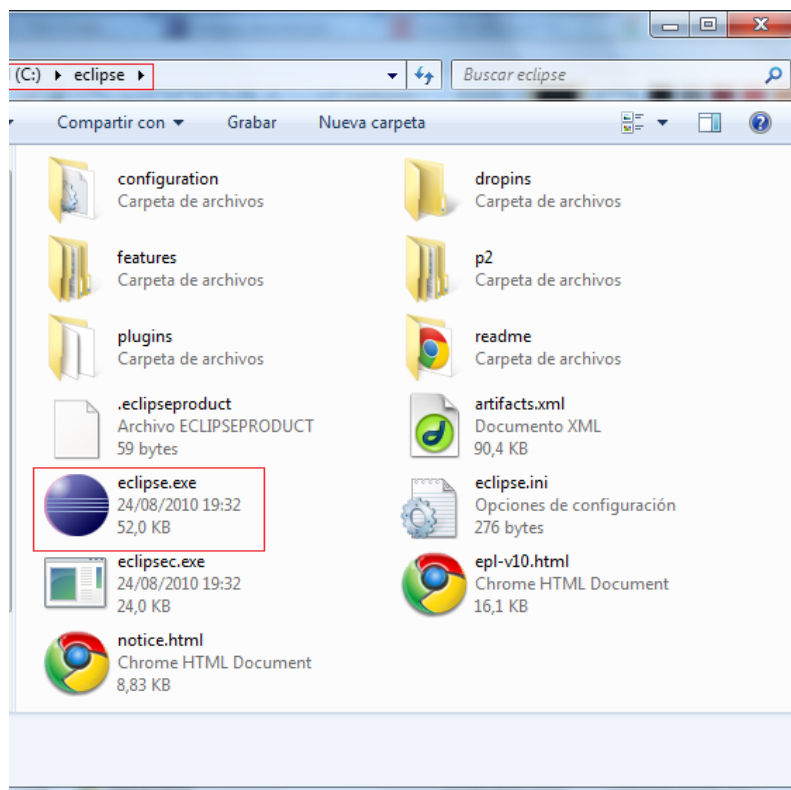


Instalación de Eclipse

Descomprimos el archivo que es un zip y no tiene un instalador.

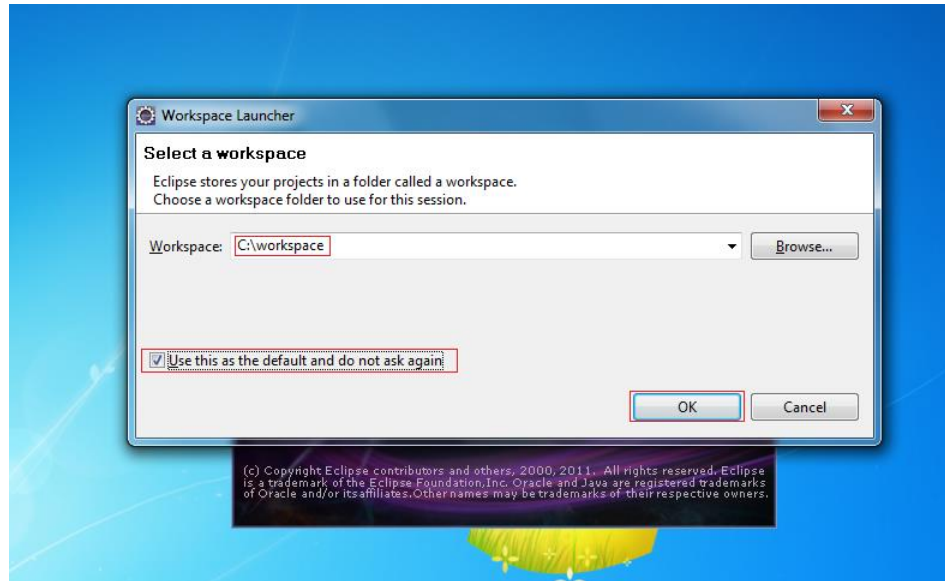


Esto se debe a que el programa no hace falta instalarlo, lo colocan donde deseen. El contenido del zip es lo siguiente:

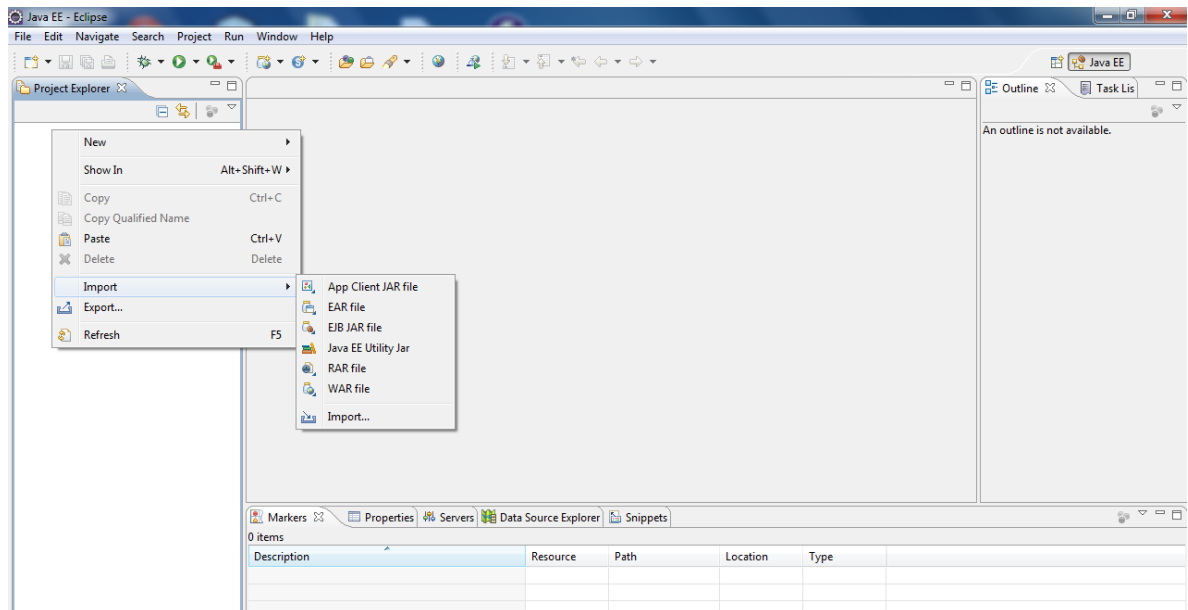


El ejecutable es el icono marcado en un cuadro rojo. Denle doble clic.

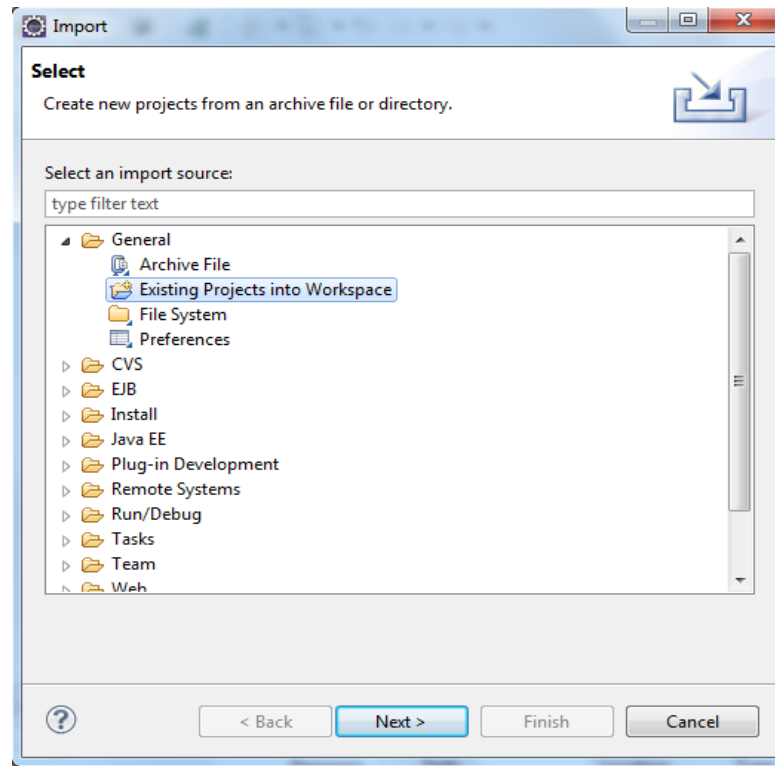
Una vez que cargue les aparecerá la siguiente pantalla donde configuraremos el workspace (área de trabajo).



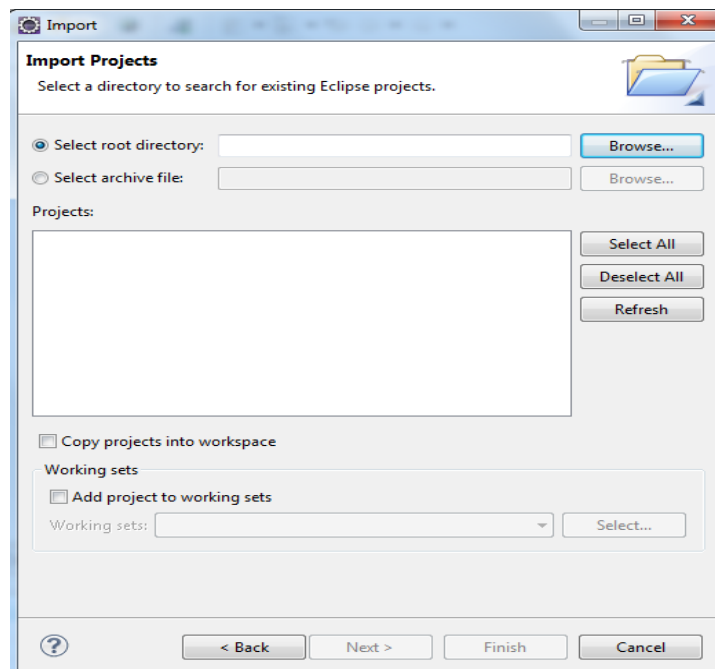
El workspace es la carpeta donde eclipse guarda todos sus proyectos. Importamos nuestro proyecto (Sistema de Monitoreo de Tesis). Hacemos clic derecho seleccionamos Import

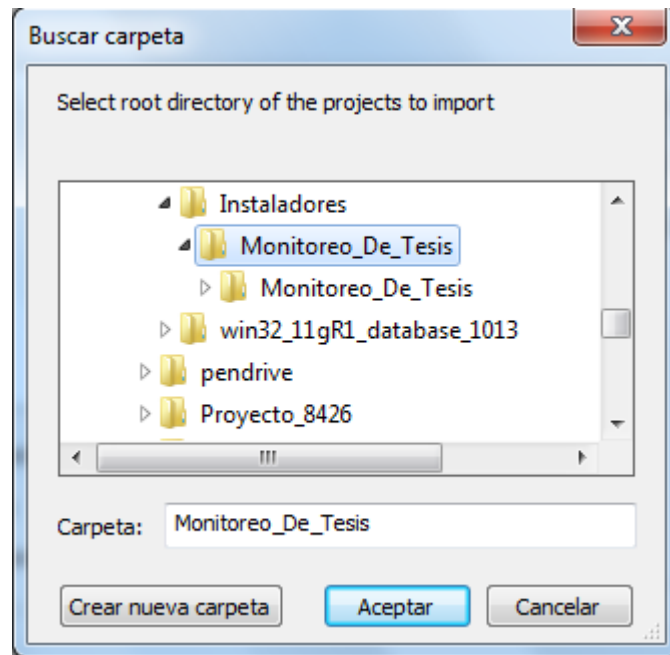


Se mostrará la siguiente pantalla seleccionamos Existing Projects into Workspace.

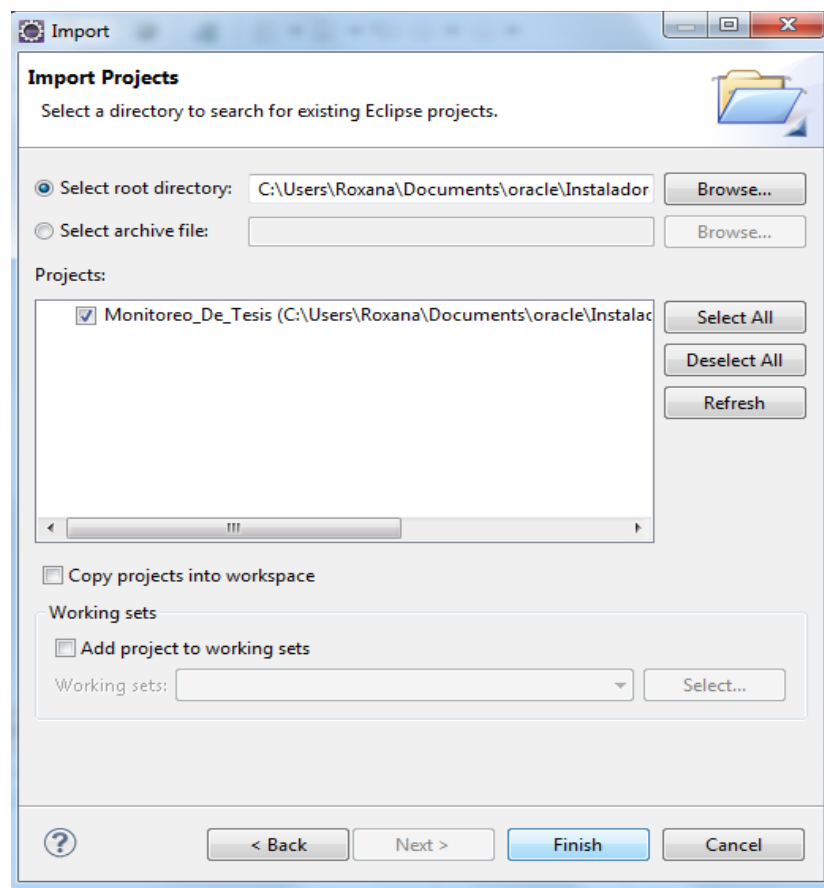


Seleccionamos browse en la siguiente pantalla para seleccionar la ruta en donde se encuentra el proyecto.





Luego le damos clic en el botón Finish para importar el proyecto.



MANUAL DE USUARIO

El SMT – Web Server es un Sistema de Monitoreo de Tesis en el cual vamos a poder llevar tanto el control de los usuarios que acceden, así como el control y seguimiento de las tesis en desarrollo y sus respectivas prorrogas; también se podrán visualizar reportes necesarios para el control de las mismas.

LOGIN

Esta es la primera pantalla de nuestro sistema donde los usuarios se podrán iniciar sesión.

Existen 3 tipos de usuarios que podrán utilizar el sistema.

1. **Director.-** Correspondiente al director de carrera en este caso exclusivamente al área de sistemas que es al cual va dirigido.

Usuario: DIRECTOR

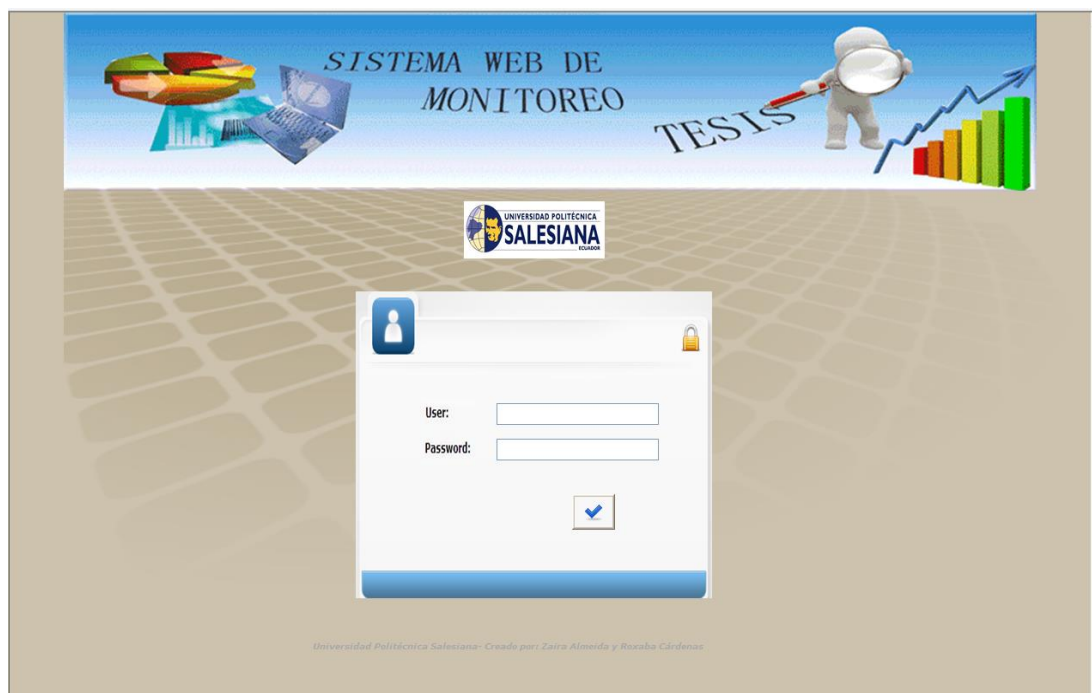
2. **Secretaria.-** Donde una encargada del departamento de secretaría de la universidad tendrá la labor y responsabilidad de hacerle el respectivo seguimiento a los usuarios y prorrogas desde este sistema.

Usuario: SECRETARIA

3. **Tutores.-** Donde solo podrán acceder los que hayan sido previamente registrados por la encargada del departamento de secretaría.

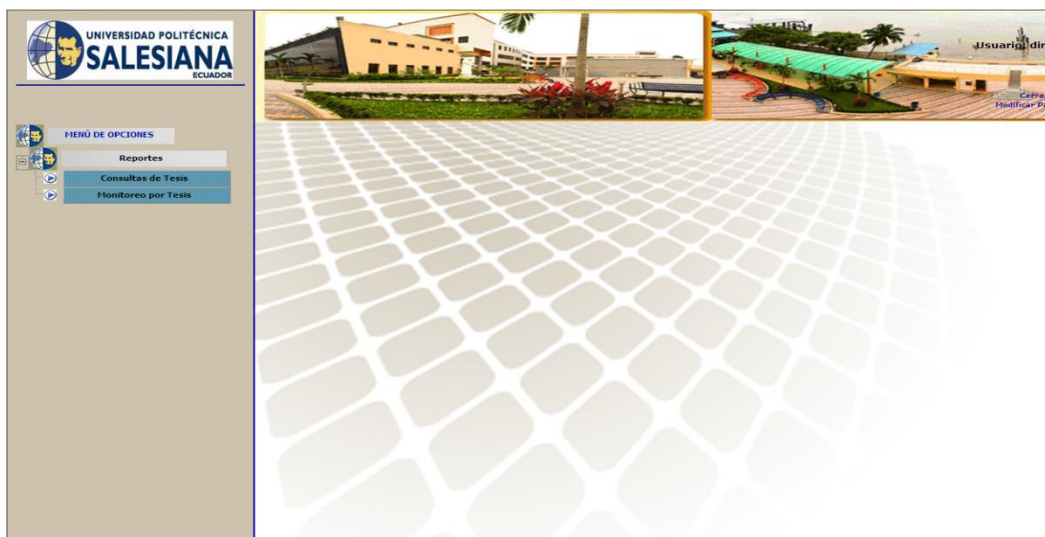
Usuario: 1letraprimernombre**primerapellido**1letrasegundoapellido

Ejemplo. RNARANJOS



A continuación detallamos las opciones del sistema para cada perfil de usuario:

DIRECTOR DE CARRERA



Pantalla de inicio cuando se ingresa como director, donde podemos visualizar una pestaña de reportes la cual a su vez tiene 2 opciones detalladas a continuación:

REPORTES



CONSULTA DE TESIS

En la siguiente opción nos aparece una ventana en la cual podemos realizar consulta de las tesis, valiéndonos de los filtros que aparecen, aquí podemos ya sea consultar sin filtros si deseamos que venga toda la información o consultar filtrando ciertos datos, así como hacer un filtro completo.

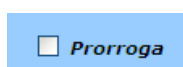
Filtro por carrera.- Se elige la carrera a la cual pertenece el Trabajo de Grado a consultarse, es el único filtro obligatorio como se indica en la pantalla.

Filtro por Tutor.- Luego de seleccionar la carrera se puede escoger un tutor de todos los que pertenecen a la carrera seleccionada con anterioridad.

Filtro por estado.- Aquí se podrá filtrar dependiendo del estado en que se encuentren la tesis, ya sea terminadas, en curso o anuladas.

A blue rectangular button with a white border. On the left, the word "Estado" is written in bold. To its right is a white text input field containing the text "Seleccione Uno". On the far right of the button is a small blue downward-pointing arrow icon.

Filtro Prorrogas.- Si se selecciona esta opción nos filtrara solo las tesis que se encuentren con prorrogas al momento de la consulta.

A blue rectangular button. On the left is a small white square checkbox. To its right, the word "Prorroga" is written in bold.

Botón Buscar.- Se presiona para realizar la consulta una vez seleccionados los filtros.



Una vez presionado el botón buscar nos aparecerá el reporte en un pdf, donde tendremos todos los datos que se necesitan.

FILTRO MONITOREO POR TESIS

En esta opción nos aparece una ventana en la cual podemos realizar la consulta del seguimiento de las tesis, sus porcentajes y capítulos de avances, consta de 2 filtros uno por carrera y por tesis.



Filtro por carrera.- Se elige la carrera a la cual pertenece el Trabajo de Grado a consultarse.

Carrera

Filtro por Tesis.- Luego de seleccionar la carrera se puede escoger una tesis de todos los que pertenecen a la carrera seleccionada con anterioridad.

Tesis

Botón Buscar.- Se presiona para realizar la consulta una vez seleccionados los filtros.



Una vez presionado el botón buscar nos aparecerá el reporte en un pdf, donde tendremos todos los datos que se necesitan.

SECRETARIA



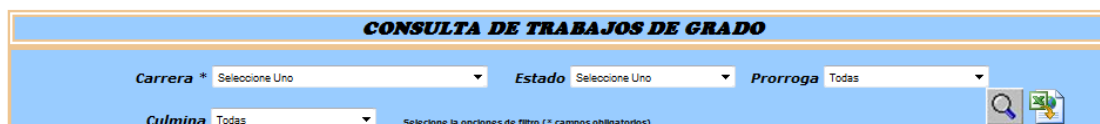
Pantalla de inicio cuando se ingresa como secretaria, donde podemos visualizar tres pestañas de reportes, usuarios, prorrogas, detalladas a continuación:

REPORTES

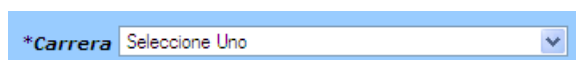


CONSULTA DE TESIS

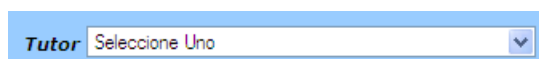
En la siguiente opción nos aparece una ventana en la cual podemos realizar consulta de las tesis, valiéndonos de los filtros que aparecen, aquí podemos ya sea consultar sin filtros si deseamos que venga toda la información o consultar filtrando ciertos datos, así como hacer un filtro completo.



Filtro por carrera.- Se elige la carrera a la cual pertenece el Trabajo de Grado a consultarse, es el único filtro obligatorio como se indica en la pantalla.



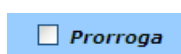
Filtro por Tutor.- Luego de seleccionar la carrera se puede escoger un tutor de todos los que pertenecen a la carrera seleccionada con anterioridad.



Filtro por estado.- Aquí se podrá filtrar dependiendo del estado en que se encuentren la tesis, ya sea terminadas, en curso o anuladas.



Filtro Prorrogas.- Si se selecciona esta opción nos filtrara solo las tesis que se encuentren con prorrogas al momento de la consulta.



Botón Buscar.- Se presiona para realizar la consulta una vez seleccionados los filtros.



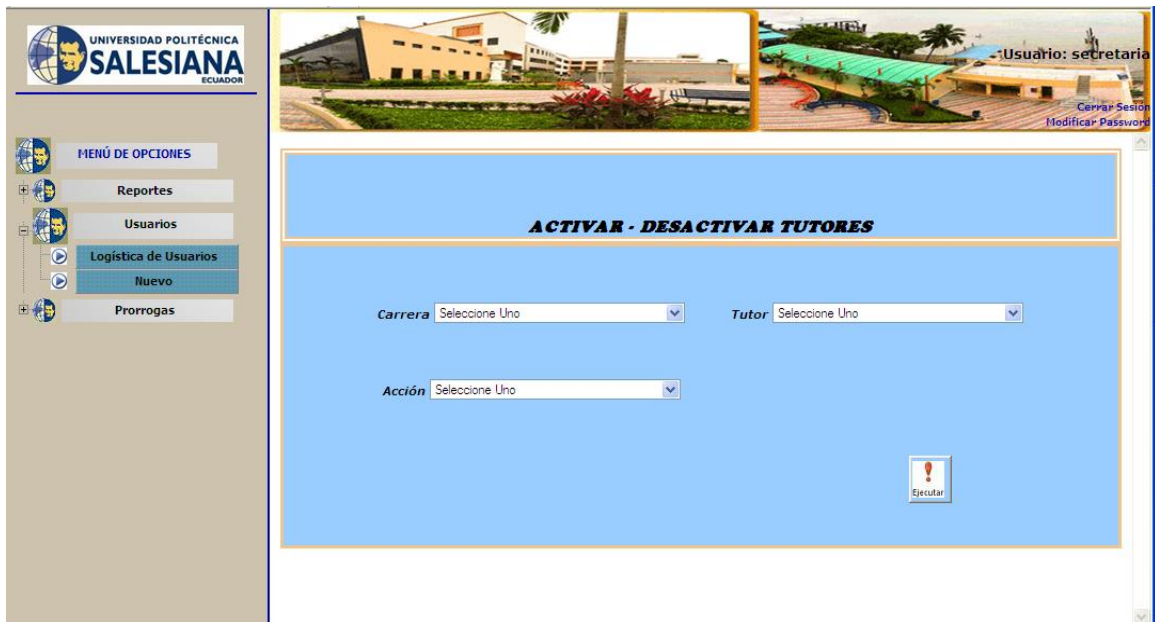
Una vez presionado el botón buscar nos aparecerá el reporte en un pdf, donde tendremos todos los datos que se necesitan.

USUARIOS



LOGÍSTICA DE USUARIOS

Al elegir esta opción aparecerá una ventana en la cual se podrá realizar tanto eliminación de usuarios que se encontraban activos, así como activación de un usuario que ya haya tenido cuenta en el sistema pero se lo haya eliminado, es decir que si no está disponible en el ingreso probablemente se encuentre en esta opción.



Filtro por carrera.- Se elige la carrera a la cual pertenece el usuario a modificar.

Carrera

Filtro por Tutor.- Luego de seleccionar la carrera se escoge el nombre del tutor al cual se desea modificar su estado de usuario.

Tutor

Filtro por estado.- Se selecciona el estado al cual se desea cambiar el usuario, ya sea eliminarlo (desactivarlo) o activarlo en caso de que se encuentra inhabilitado al haber sido anulado anteriormente.

Estado

Botón Ejecutar.- Se presiona para realizar la modificación al usuario seleccionado.



NUEVO

Al presionar esta opción del menú nos aparecerá una ventana en la cual podremos ingresar un nuevo usuario, en esta opción solo se ingresarán tutores que no hayan tenido una cuenta en el sistema.

La imagen muestra una interfaz web de la Universidad Politécnica Salesiana. A la izquierda hay un menú de opciones con botones para Reportes, Usuarios, Logística de Usuarios (seleccionado), Nuevo y Prorrogas. El área principal tiene un encabezado con una imagen de un campus y el título "INGRESO DE NUEVOS USUARIOS/TUTORES". Debajo, hay dos campos de selección: "Carrera" y "N. Tutor", ambos con el texto "Seleccione Uno". Al final, hay un botón con una marca de verificación azul. En la esquina superior derecha, se puede ver parte de un menú de usuario con opciones como "Cerrar Sesión" y "Modificar Password".

Filtro por carrera.- Se elige la carrera a la cual pertenece el usuario tutor a ingresar.

Un campo de selección con el label "Carrera" a la izquierda. El menú desplegable muestra "Seleccione Uno" y un icono de flecha hacia abajo.

Filtro por Tutor.- Luego de seleccionar la carrera se escoge el nombre del tutor al cual se desea ingresar como nuevo usuario.

Un campo de selección con el label "Tutor" a la izquierda. El menú desplegable muestra "Seleccione Uno" y un icono de flecha hacia abajo.

Botón aceptar.- Se presiona para guardar.



PRORROGAS



INGRESOS

Al seleccionar esta opción del menú podremos ingresar las prorrogas que han sido aprobadas para cada Trabajo de Grado.

A screenshot of the 'SISTEMA WEB DE MONITOREO TESIS' interface. On the left is a sidebar with the 'UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR' logo and a menu with options: 'Reportes', 'Usuarios', 'Prorrogas', and 'Ingresos'. The main content area has a header banner with the title 'SISTEMA WEB DE MONITOREO TESIS' and a graphic of a person with a magnifying glass. Below the banner is a section titled 'INGRESO DE PRORROGAS' containing a form with a 'Carrera' dropdown menu (showing 'Seleccione Uno') and a 'Cedula' text input field with a search icon.

Filtro por carrera.- Se elige la carrera a la cual pertenece el trabajo de grado.

A close-up of the 'Carrera' dropdown menu from the 'INGRESO DE PRORROGAS' form. It shows the text 'Carrera' followed by a dropdown box containing 'Seleccione Uno' and a downward arrow.

Filtro por cédula de estudiante.- Se ingresa el número de cédula del estudiante que está realizando la tesis, en caso de ser 2 alumnos se ingresa el de uno.

Cedula

Botón Buscar.- Se presiona para realizar la consulta una vez seleccionados los filtros.



Una vez que se ha presionado el botón buscar en la parte inferior de la cabecera de consulta aparecerán los datos del Trabajo de Grado así como sus respectivos estudiantes que lo están desarrollando.

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

USUARIO: secretaria

Cerrar Sesión

Modificar Password

INGRESO DE PRORROGAS

Carrera: INGENIERÍA DE SISTEMAS Cedula: 0926799578

Tesis :

"ESTUDIO Y AUTOMATIZACIÓN DEL PROCESO DE CONTROL Y MONITOREO DE ELABORACIÓN DE TESIS DE ESTUDIANTES DE LA CARRERA DE INGENIERÍA DE SISTEMAS"

Alumnos:

926799578 - ALMEIDA MURILLO ZAIRA JANETH

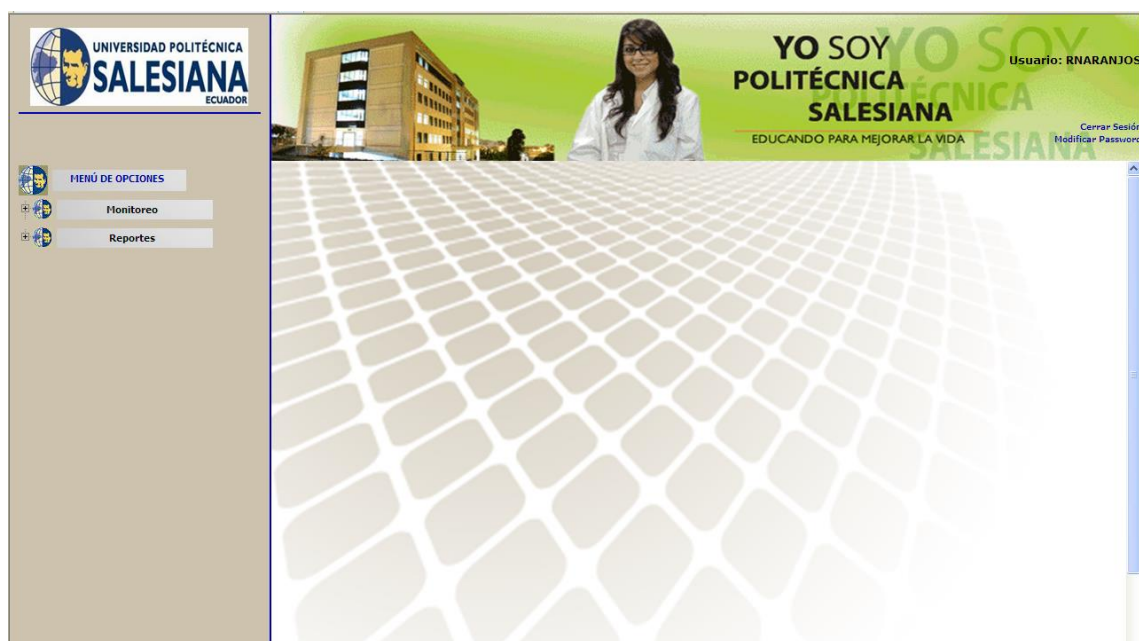
927523092 - CARDENAS HERNANDEZ ROXANA ESTEFANIA

Guardar

Al presionar el botón guardar se almacenarán la prórroga para esa tesis.

Guardar

TUTORES



Pantalla de inicio cuando se ingresa como usuario tutor, donde podemos visualizar una pestaña de monitoreo y una de reportes las cuales a su vez tienen opciones detalladas a continuación:

MONITOREO



LOGÍSTICA DE SEGUIMIENTO

En esta opción nos aparece una ventana en la cual podemos realizar el ingreso de seguimiento de las tesis, sus capítulos de avances, consta de 2 filtros uno por carrera y por tesis.



Filtro por carrera.- Se elige la carrera a la cual pertenece el Trabajo de Grado a consultarse.

Carrera

Filtro por Tesis.- Luego de seleccionar la carrera se escoge un tema de tesis de todos los que pertenecen a la carrera seleccionada con anterioridad y que pertenecen a su usuario tutor con el que esta logueado.

Tesis

Botón Buscar.- Se presiona para realizar la consulta una vez seleccionados los filtros.



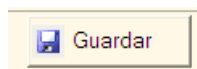
Una vez presionado el botón buscar si es la primera vez que se va a ingresar seguimiento de una tesis aparecerá un formulario como este:

CAPITULOS	OBSERVACIÓN	ESTADO	Fecha
ANTECEDENTES	OBSERVACIÓN	Pendiente	17/04/2013
CAPITULO1			
CAPITULO2			
CAPITULO3			

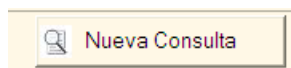
Pero si ya se encuentra seguimiento de esta tesis le aparecerá una ventana como esta:

CAPITULOS	OBSERVACIÓN	ESTADO	Fecha
ANTECEDENTES	OBSERVACIÓN	Pendiente	17/04/2013
CAPITULO1			
PROBLEMA INVESTIGACION	finalizado ya queremos ustenar y graduarnos este año x favor dejenos graduuuuuuaarr	FINALIZADO	15-ABR-13
PROBLEMA INVESTIGACION	no termino	PENDIENTE	15-ABR-13
CAPITULO2			
CAPITULO3			
PRUEBAS Y RESULTADOS	terminado	FINALIZADO	15-ABR-13

En ambos casos se deberá seleccionar los subcapítulos que se encuentren realizados y luego presionamos el botón guardar.



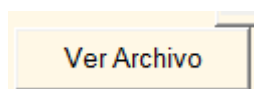
Para cancelar podemos presionar el botón Nueva Consulta el cual nos blanqueará el formulario.



Para subir un documento de tesis podemos presionar el botón de Cargar Archivo nos aparecerá una ventanita para examinar donde se encuentra nuestro archivo.



Para visualizar el archivo anteriormente cargado se le dará clic en el botón de ver archivo.



REPORTES

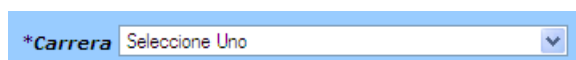


CONSULTA DE TESIS

En la siguiente opción nos aparece una ventana en la cual podemos realizar consulta de las tesis, valiéndonos de los filtros que aparecen, aquí podemos ya sea consultar sin filtros si deseamos que venga toda la información o consultar filtrando ciertos datos, así como hacer un filtro completo.



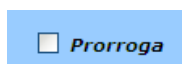
Filtro por carrera.- Se elige la carrera a la cual pertenece el Trabajo de Grado a consultarse, es el único filtro obligatorio como se indica en la pantalla.



Filtro por estado.- Aquí se podrá filtrar dependiendo del estado en que se encuentren la tesis, ya sea terminadas, en curso o anuladas.



Filtro Prorrogas.- Si se selecciona esta opción nos filtrara solo las tesis que se encuentren con prorrogas al momento de la consulta.



Botón Buscar.- Se presiona para realizar la consulta una vez seleccionados los filtros.



Una vez presionado el botón buscar nos aparecerá el reporte en un pdf, donde tendremos todos los datos que se necesitan.

FILTRO MONITOREO POR TESIS

En esta opción nos aparece una ventana en la cual podemos realizar la consulta del seguimiento de las tesis, sus porcentajes y capítulos de avances, consta de 2 filtros uno por carrera y por tesis.



Filtro por carrera.- Se elige la carrera a la cual pertenece el Trabajo de Grado a consultarse.

Carrera

Filtro por Tesis.- Luego de seleccionar la carrera se puede escoger una tesis de todos los que pertenecen a la carrera seleccionada con anterioridad.

Tesis

Botón Buscar.- Se presiona para realizar la consulta una vez seleccionados los filtros.

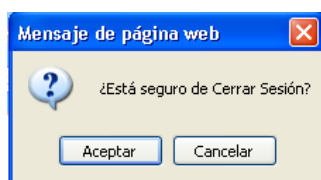


Una vez presionado el botón buscar nos aparecerá el reporte en un pdf, donde tendremos todos los datos que se necesitan.

CERRAR SESIÓN



Al presionar esta opción que está presente en todas las ventanas en la parte superior derecha se podrá cerrar la sesión de usuario, previa confirmación del mensaje emergente que aparecerá.



MODIFICAR PASSWORD



Al presionar esta opción que está presente en todas las ventanas en la parte superior derecha se podrá cambiar la clave del usuario logueado, aparecerá una ventana en la que deberá ingresar la clave actual y dos veces la clave nueva.

Un formulario con el título "Modificar Password". Contiene los siguientes campos: "Usuario:" con el valor "director", "Password Actual:" (campo vacío), "Nuevo Password:" (campo vacío) con el texto "(máximo 15 caracteres)" debajo, y "Confirmar Password:" (campo vacío). Hay un botón de confirmación con un icono de checkmark.